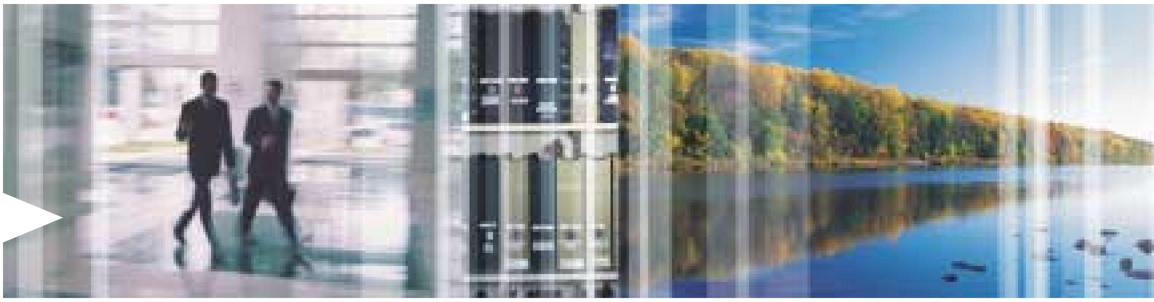
CONTROL-O®/COSMOS

User Guide



Supporting

CONTROL-O® version 6.2.18

July 31, 2006

Contacting BMC Software

You can access the BMC Software website at http://www.bmc.com. From this website, you can obtain information about the company, its products, corporate offices, special events, and career opportunities.

United States and Canada

Address	BMC SOFTWARE INC 2101 CITYWEST BLVD HOUSTON TX 77042-2827 USA	**Telephone**	713 918 8800 or 800 841 2031	**Fax**	713 918 8000

Outside United States and Canada

Telephone	(01) 713 918 8800	**Fax**	(01) 713 918 8000

Copyright 2006 BMC Software, Inc., as an unpublished work. All rights reserved.

BMC Software, the BMC Software logos, and all other BMC Software product or service names are registered trademarks or trademarks of BMC Software, Inc.

DB2 is a registered trademark of International Business Machines Corporation.

All other trademarks belong to their respective companies.

BMC Software considers information included in this documentation to be proprietary and confidential. Your use of this information is subject to the terms and conditions of the applicable End User License Agreement for the product and the proprietary and restricted rights notices included in this documentation.

Restricted rights legend

U.S. Government Restricted Rights to Computer Software. UNPUBLISHED -- RIGHTS RESERVED UNDER THE COPYRIGHT LAWS OF THE UNITED STATES. Use, duplication, or disclosure of any data and computer software by the U.S. Government is subject to restrictions, as applicable, set forth in FAR Section 52.227-14, DFARS 252.227-7013, DFARS 252.227-7014, DFARS 252.227-7015, and DFARS 252.227-7025, as amended from time to time. Contractor/Manufacturer is BMC SOFTWARE INC, 2101 CITYWEST BLVD, HOUSTON TX 77042-2827, USA. Any contract notices should be sent to this address.

Customer support

You can obtain technical support by using the Support page on the BMC Software website or by contacting Customer Support by telephone or e-mail. To expedite your inquiry, please see "Before Contacting BMC Software."

Support website

You can obtain technical support from BMC Software 24 hours a day, 7 days a week at http://www.bmc.com/support_home. From this website, you can

- read overviews about support services and programs that BMC Software offers
- find the most current information about BMC Software products
- search a database for problems similar to yours and possible solutions
- order or download product documentation
- report a problem or ask a question
- subscribe to receive e-mail notices when new product versions are released
- find worldwide BMC Software support center locations and contact information, including e-mail addresses, fax numbers, and telephone numbers

Support by telephone or e-mail

In the United States and Canada, if you need technical support and do not have access to the web, call 800 537 1813 or send an e-mail message to support@bmc.com. Outside the United States and Canada, contact your local support center for assistance.

Before contacting BMC Software

Before you contact BMC Software, have the following information available so that Customer Support can begin working on your problem immediately:

- product information
  - product name
  - product version (release number)
  - license number and password (trial or permanent)
- operating system and environment information
  - machine type
  - operating system type, version, and service pack or other maintenance level such as PUT or PTF
  - system hardware configuration
  - serial numbers
  - related software (database, application, and communication) including type, version, and service pack or maintenance level
- sequence of events leading to the problem
- commands and options that you used
- messages received (and the time and date that you received them)
  - product error messages
  - messages from the operating system, such as `file system full`
  - messages from related software

Contents

Figures

Tables

About This Guide

The CONTROL-O/COSMOS User Guide is the main publication that describes CONTROL-O/COSMOS software. It contains detailed information on all CONTROL-O/COSMOS functions and facilities, and is comprised of the following chapters:

Chapter 1–Introduction to CONTROL-O/COSMOS

Brief introduction to the facilities and features of CONTROL-O/COSMOS.

Chapter 2–Online Facilities

Guide to the CONTROL-O/COSMOS Online facilities. All CONTROL-O/COSMOS screens are displayed and discussed in logical sequence.

Chapter 3–Component Details

Detailed description of all components of CONTROL-O/COSMOS and how they work together to manage objects at your site.

Chapter 4–Implementation

Detailed description of how to test and implement CONTROL-O/COSMOS at your site.

Conventions Used in This Guide

Notational conventions that may be used in this guide are explained below.

Standard Keyboard Keys

Keys that appear on the standard keyboard are identified in boldface, for example, **Enter, Shift, Ctrl+S** (a key combination), or **Ctrl S** (a key sequence).

WARNING

The commands, instructions, procedures, and syntax illustrated in this guide presume that the keyboards at your site are mapped in accordance with the EBCDIC character set. Certain special characters are referred to in this documentation, and you must ensure that your keyboard enables you to generate accurate EBCDIC hex codes. This is particularly true on keyboards that have been adapted to show local or national symbols. You should verify that

- **$** is mapped to x'5B
- **#** is mapped to x'7B'
- **@** is mapped to x'7C'

If you have any questions about whether your keyboard is properly mapped, contact your system administrator.

Preconfigured PFKeys

Many commands are preconfigured to specific keys or key combinations. This is particularly true with regard to numbered PF keys, or pairs of numbered PFKeys. For example, the END command is preconfigured to, and indicated as, **PF03/PF15**. To execute the END command, press either the **PF03** key or the **PF15** key.

Instructions to enter commands may include

- only the name of the command, such as, enter the END command
- only the PF keys, such as, press **PF03/PF15**
- or both, such as, press **PF03/PF15,** or enter the END command

Command Lines and Option Fields

Most screens contain a command line, which is primarily used to identify a single field where commands, or options, or both, are to be entered. These fields are usually designated COMMAND, but they are occasionally identified as COMMAND / OPT or COMMAND / OPTION.

Option field headings appear in many screens. These headings sometimes appear in the screen examples as OPTION, or OPT, or O.

Names of Commands, Fields, Files, Functions, Jobs, Libraries, Members, Missions, Options, Parameters, Reports, Subparameters, and Users

The names of commands, fields, functions, jobs, libraries, members, missions, options, parameters, reports, subparameters, users, and most files, are shown in standard UPPERCASE font.

User Entries

In situations where you are instructed to enter characters using the keyboard, the specific characters to be entered are shown in this **UPPERCASE BOLD** text, for example, type **EXITNAME.**

Syntax statements

In syntax, the following additional conventions apply:

- A vertical bar (|) separating items indicates that you must choose one item. In the following example, you would choose *a*, *b*, or *c*:

  ```
  a | b | c
  ```

- An ellipsis (. . .) indicates that you can repeat the preceding item or items as many times as necessary.
- Square brackets ([ ]) around an item indicate that the item is optional. If square brackets ([ ]) are around a group of items, this indicates that the item is optional, and you may choose to implement any single item in the group. Square brackets can open ([ ) and close ( ]) on the same line of text, or may begin on one line of text and end, with the choices being stacked, one or more lines later.
- Braces ({ }) around a group of items indicates that the item is mandatory, and you must choose to implement a single item in the group. Braces can open ({) and close (}) on the same line of text, or may begin on one line of text and end, with the choices being stacked, one or more lines later.

Screen Characters

`All syntax, operating system terms, and literal examples are presented in this typeface.` This includes JCL calls, code examples, control statements, and system messages. Examples of this are:

- calls, such as

```
CALL ’CBLTDLI’
```

- code examples, such as

```
FOR TABLE owner.name USE option, . . . ;
```

- control statements, such as

```
//PRDSYSIN DD * USERLOAD PRD(2) PRINT
```

- system messages, both stand-alone, such as `You are not logged on to database database_name`, and those embedded in text, such as the message `You are not logged on to database database_name`, are displayed on the screen.

Variables

Variables are identified with *italic* text. Examples of this are:

- In syntax or message text, such as
  `Specify database` *database_name*
- In regular text, such as
  replace database *database_name1* with database *database_name2* for the current session
- In a version number, such as
  EXTENDED BUFFER MANAGER for IMS 4.1.*xx*

Special elements

This book includes special elements called *notes* and *warnings*:

NOTE

Notes provide additional information about the current subject.

WARNING

Warnings alert you to situations that can cause problems, such as loss of data, if you do not follow instructions carefully.

Information New to This Version

Where substantive additions and modifications to the content of this guide occur, revision bars have been inserted in the margin.

Additional information that is new to this version is described in Appendix C of the *INCONTROL Upgrade Guide*.

Related Publications

CONTROL-O Conversion Guide

Concepts and steps required for converting from other console automation products to CONTROL-O.

CONTROL-O SolveWare Reference Guide

Predefined solutions and rules that can be used for solving commonly encountered automation problems.

CONTROL-O User Guide

A complete guide to CONTROL-O features, options, and usage.

CONTROL-O/Server User Guide

Complete guide to CONTROL-O / Server features, options and usage.

INCONTROL for z/OS Administrator Guide

Information for system administrators about customizing and maintaining INCONTROL™ family of products.

INCONTROL for z/OS Installation Guide

Step-by-step guide to installing INCONTROL family of products using the INCONTROL Installation and Customization Engine™ (ICE) application.

INCONTROL for z/OS Messages Manual

Comprehensive listing and explanation of all INCONTROL and IOA messages and codes.

INCONTROL for z/OS Security Guide

Step-by-step guide to implementing security in INCONTROL family of products using the ICE application.

INCONTROL for z/OS Utilities Guide

Describes utilities designed to perform specific administrative tasks that are available to INCONTROL products.

Chapter 1

Introduction to CONTROL-O/COSMOS

This chapter includes the following topics:

Introduction

CONTROL-O / COSMOS monitors objects (started tasks, terminals, disks, and so on) in your computing environment, and maintains each object in a status specified by the user. If the desired status of an object changes, CONTROL-O / COSMOS performs the actions necessary to bring the object to its new desired status.

For example, if CICS is up but it becomes necessary to bring CICS down, the desired status of CICS can be changed to DOWN and CONTROL-O / COSMOS takes the steps necessary to bring CICS down.

Object Status

Two types of status are relevant for each CONTROL-O / COSMOS-controlled object:

- Current status – Current status of the object.
- Desired status – Preferred status of the object.

When the desired status and the current status match (for example, when both the current status and the desired status of object CICS are UP), no changes are necessary regarding the object.

When the desired status and current status of an object do not match (for example, the current status of the CICS object is DOWN and the desired status is UP), CONTROL-O / COSMOS detects this and proceeds with actions designed to bring the object to the desired status.

Desired status of objects is set initially according to an option chosen by the user during CONTROL-O / COSMOS startup. The user can then modify the desired status using:

- An option in the CONTROL-O / COSMOS Object Status screen, described in "CONTROL-O / COSMOS Object Status Screen" on page 29.

- CONTROL-O / COSMOS command COSCMD, described in "COSCMD Command" on page 81.

Direct user intervention is not necessarily required to change the desired status of an object. Use of CONTROL-O rules can often replace direct intervention. Once the appropriate rule is defined, the changing of the desired status of the object can be initiated by the rule.

Component Overview

The following concepts and components are central to CONTROL-O / COSMOS functionality:

Table 1 CONTROL-O/COSMOS Concepts

Concept	Description
Databases	A set of AutoEdit variable databases that store information for managing CONTROL-O / COSMOS-controlled objects.
Rules	CONTROL-O rules specifically designed to manage CONTROL-O / COSMOS databases.
Commands	Commands that affect CONTROL-O / COSMOS operations and the information in CONTROL-O / COSMOS databases.
Screens	Screens that enable the user to view and modify CONTROL-O / COSMOS information.

These components are described briefly in this section, and in detail in Chapter 3, “Component Details.”

CONTROL-O/COSMOS Databases

The following AutoEdit databases are used by CONTROL-O / COSMOS:

- Object Databases
- Prerequisite Database
- Method Databases

NOTE

For information on managing several machines within the same Sysplex, see “Using CONTROL-O / COSMOS to Manage a Sysplex” on page 90.

Object Databases

As mentioned earlier, CONTROL-O / COSMOS monitors objects (started tasks, I / O devices, and so on) in the environment and maintains those objects in their specified desired status. It is therefore necessary to provide CONTROL-O / COSMOS with a list and description of those objects. This information about CONTROL-O / COSMOS-controlled objects is stored in Object databases.

Each row in an Object database contains information about a specific object (for example, object name, class, current status, desired status and mode). Multiple Object databases can be defined.

CONTROL-O / COSMOS periodically scans the current and desired status for each object in the Object databases. If the current and desired statuses for an object do not match, CONTROL-O / COSMOS attempts to bring the object to the desired status (meaning, so that the current status matches the desired status).

Object databases must reflect your current computing environment. They can be generated manually, or they can be automatically generated using the SYSIMAGE facility, as described on page 96.

Attributes of the objects in the Object databases can be modified by the user at any time to reflect desired changes in the environment or changes that have already occurred.

Prerequisite Database

In many cases, prerequisite criteria must be satisfied before the status of an object is changed (for example, VTAM must be up before CICS can be activated).

It is therefore necessary to provide CONTROL-O / COSMOS with appropriate prerequisite information so that CONTROL-O / COSMOS can ensure that prerequisites are satisfied before changing the status of an object. This prerequisite information is stored in the Prerequisite database. Each row in the Prerequisite database describes an object that must be up before another specified object is brought up.

Before attempting to change the (current) status of an object, CONTROL-O / COSMOS checks the Prerequisite database for prerequisite criteria that must be satisfied.

If all prerequisites for changing the object’s status are satisfied (or if there are no prerequisites), CONTROL-O / COSMOS triggers a rule that modifies the object’s status.

If certain prerequisites are not satisfied, the change is not performed at that time (for example, if VTAM is not up, CICS is not started). CONTROL-O / COSMOS, however, continues to scan the Object and Prerequisite databases. This is important because environments are dynamic, and a prerequisite that is not satisfied at one point in time may be satisfied at a later time. (If, for example, VTAM is brought to an UP status, and therefore the prerequisites for starting CICS are satisfied, CONTROL-O / COSMOS starts CICS the next time it scans the databases.)

All objects referenced in the Prerequisite database must be CONTROL-O / COSMOS-controlled objects (meaning, objects defined in a CONTROL-O / COSMOS Object database).

Method Databases

When a change to the status of an object is to be performed (that is, when the current and desired status of the object do not match and all prerequisites are satisfied), CONTROL-O / COSMOS needs to select the appropriate CONTROL-O rule to perform the actions that bring the object to its desired status. These rules are referred to as Method rules. Information that indicates which Method rule is appropriate for performing the desired change to an object is stored in Method databases.

Each row in a Method database defines a method. A method consists of the particular type of change and the Method rule that can bring about the change. Each row (method) lists an object, object class, current status, desired status, and the name of the Method rule that is used to affect the desired change.

For example, a row in a Method database may indicate that the method defined in the row is used for modifying an object named CICSTEST with a current status of DOWN and a desired status of UP. When invoked, the Method rule specified in this row performs actions that bring the object to a status of UP.

Multiple Method databases can be defined.

CONTROL-O/COSMOS Rules

A collection of predefined rules is supplied with CONTROL-O / COSMOS. These rules detect and intercept changes in the computing environment, update the Object databases accordingly, and periodically verify the accuracy of information in the Object databases. Some types of rules supplied with CONTROL-O / COSMOS are:

- Rules that perform actions required to modify the current status of CONTROL-O / COSMOS-controlled objects (meaning, Method rules).
- Rules that intercept initialization and termination messages of common started tasks (for example, CICS, DB2 and JES2), and update the current status of these tasks in the Object database.
- Rules that intercept Start and Stop commands and set the desired status of appropriate objects to UP or DOWN instead. (CONTROL-O / COSMOS detects the change in the desired status the next time it scans the Object database, and implements Method rules specified by the user to bring the object to the desired status.)
- Rules that periodically check if the values in the Object database represent the real situation in the system.
- Rules that execute CONTROL-O / COSMOS commands, described in Chapter 3, “Component Details.”

NOTE

When implementing CONTROL-O / COSMOS to handle IPL and shutdown processes, ensure that other existing rules do not overlap. To avoid this, it is recommended that you implement CONTROL-O rules, handling IPLs, or shutdown procedures as Method rules. For more details on Method rules, see "Method Databases" on page 21.

CONTROL-O/COSMOS Commands

A variety of commands are provided with the CONTROL-O / COSMOS facility. These commands can be divided into the following groups:

- Commands that affect the operation of CONTROL-O / COSMOS.
- Commands that perform actions on CONTROL-O / COSMOS-controlled objects. Many of these commands perform actions that can also be performed using CONTROL-O / COSMOS online screens (described below).

Most CONTROL-O / COSMOS commands trigger CONTROL-O / COSMOS rules that perform desired actions. These command rules can be modified, which enable you to customize the CONTROL-O / COSMOS facility according to your preferred operating procedures (for example, to issue Shout messages describing the action performed)

CONTROL-O/COSMOS Online Screens

Online screens are provided with CONTROL-O / COSMOS. These screens are primarily used to manage ongoing operations. They are accessed using option OC of the IOA Primary Option menu. The following CONTROL-O / COSMOS screens are available:

Table 2 CONTROL-O/COSMOS Screen Types

Screen	Description
Object Status screen	Displays objects and allows actions (for example, changing operation mode) on individual objects.
Database Status screen	Displays a list of Object databases and allows actions on all objects in a specific database or multiple databases.

Options of the CONTROL-O / COSMOS online screens can also be implemented by various operator commands, described in Chapter 3, "Component Details." For more information about CONTROL-O / COSMOS online screens, see Chapter 2, "Online Facilities."

CONTROL-O/COSMOS Security

CONTROL-O / COSMOS security is handled by two security modules:

- IOASE42 – Protects the definition and administration of the IOA Global variables database. For more information, see the IOASE42 section in the *INCONTROL for z/OS Security Guide*.

- CTOSE04 – Verifies user authorization to specific CONTROL-O / COSMOS Online entities. For more information, see the CTOSE04 section in the *INCONTROL for z/OS Security Guide*.

Chapter 2

Online Facilities

Introduction

CONTROL-O / COSMOS monitors the status of specified objects (resources) in your computing environment and if necessary modifies the status of an object when its current status does not match a specified desired status.

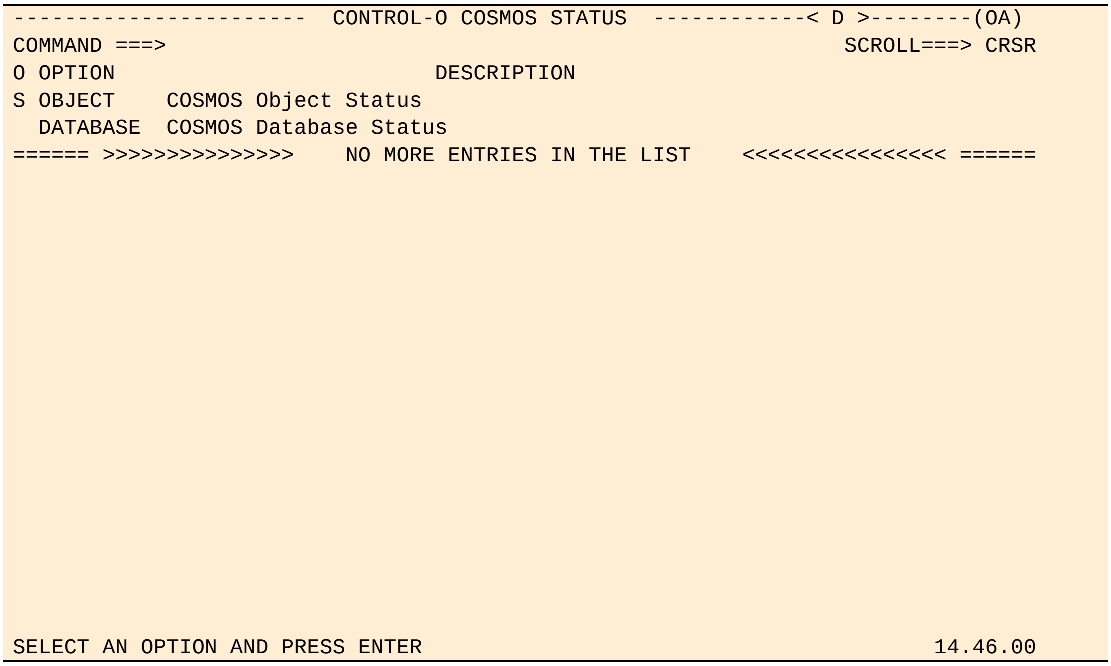
CONTROL-O / COSMOS screens facilitate management of CONTROL-O / COSMOS-controlled objects and databases of objects.

CONTROL-O/COSMOS Main Menu

To enter the CONTROL-O / COSMOS Main menu, specify option OC in the IOA Primary Option menu and press **Enter**. For more information about the IOA Primary Option menu, see the online facilities chapter of the *CONTROL-O User Guide*.

The following screen is displayed:

Figure 1 CONTROL-O/COSMOS Status Screen



```
-----------------------  CONTROL-O COSMOS STATUS  ------------< D >--------(OA)
COMMAND ===>                                                     SCROLL===> CRSR
O OPTION                         DESCRIPTION
S OBJECT    COSMOS Object Status
  DATABASE  COSMOS Database Status
====== >>>>>>>>>>>>>>>>    NO MORE ENTRIES IN THE LIST    <<<<<<<<<<<<<<<< ======















SELECT AN OPTION AND PRESS ENTER                                        14.46.00
```

This screen displays the available CONTROL-O / COSMOS screens. The following screens are available:

Table 3 Status Screens

Screen	Description
OBJECT	Object Status screen. This screen allows management of individual CONTROL-O / COSMOS-controlled objects.
DATABASE	Database Status screen. This screen allows management of CONTROL-O / COSMOS-controlled objects in a specific database or multiple databases.

OS databases. Options specified in this screen affect the functioning of all objects in the specified database or databases.

Select a CONTROL-O / COSMOS screen by specifying **S** (Select) to the left of the desired option and pressing **ENTER**.

To return to the IOA Primary Option menu, press **END** (**PF03/PF15**).

NOTE

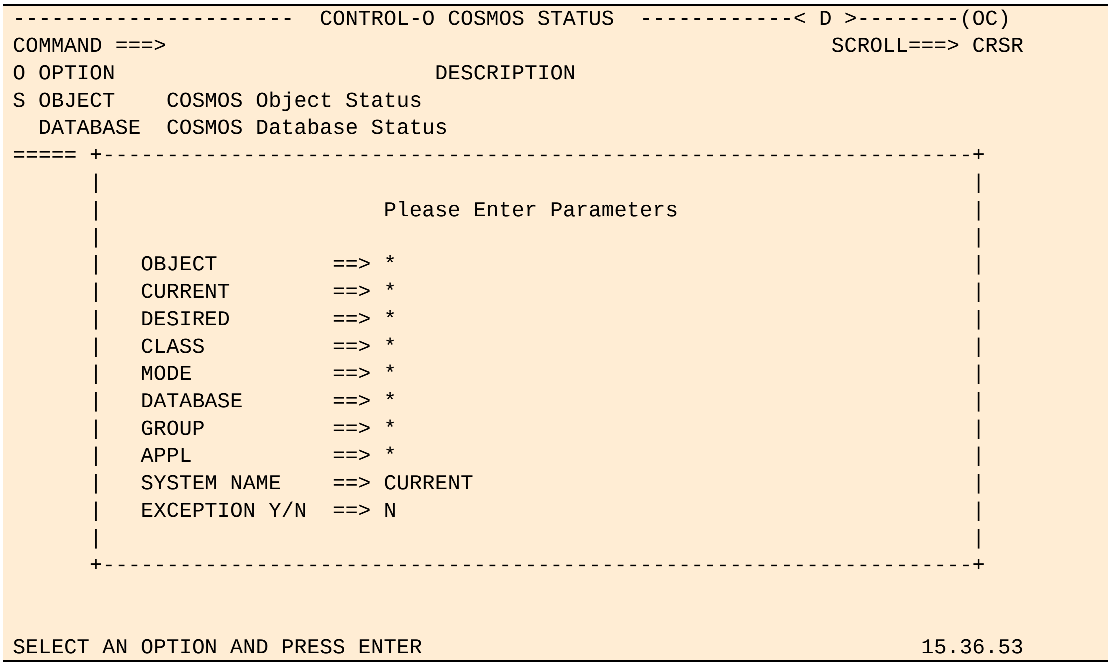
During startup or shutdown of the system, the CONTROL-O / COSMOS screens may not be available. For this reason, a set of operator commands can be used to perform actions normally performed using CONTROL-O / COSMOS screens. For more information about CONTROL-O / COSMOS commands, see Chapter 3, “Component Details.”

CONTROL-O/COSMOS Object Status Screen – Selection Criteria

The CONTROL-O / COSMOS Object Status screen enables management of CONTROL-O / COSMOS-controlled objects on an individual basis. For example, this screen can be used to view object status or to change the CONTROL-O / COSMOS mode for an object.

To display the Object Status screen, specify **S** (Select) to the left of option OBJECT in the CONTROL-O / COSMOS Main menu and press **ENTER**. The following Selection Criteria window is displayed:

Figure 2 Object Status Screen



```
----------------------  CONTROL-O COSMOS STATUS  ------------< D >--------(OC)
COMMAND ===>                                                     SCROLL===> CRSR
O OPTION                         DESCRIPTION
S OBJECT    COSMOS Object Status
  DATABASE  COSMOS Database Status
===== +---------------------------------------------------------------------+
      |                                                                     |
      |                      Please Enter Parameters                        |
      |                                                                     |
      |   OBJECT         ==> *                                              |
      |   CURRENT        ==> *                                              |
      |   DESIRED        ==> *                                              |
      |   CLASS          ==> *                                              |
      |   MODE           ==> *                                              |
      |   DATABASE       ==> *                                              |
      |   GROUP          ==> *                                              |
      |   APPL           ==> *                                              |
      |   SYSTEM NAME    ==> CURRENT                                        |
      |   EXCEPTION Y/N  ==> N                                              |
      |                                                                     |
      +---------------------------------------------------------------------+


SELECT AN OPTION AND PRESS ENTER                                        15.36.53
```

The Selection Criteria window enables the filtering of objects displayed in the Objects Status screen.

When you first open the window, most of the criteria have an * value. Filtering is not performed on criteria that have* as their value. Therefore, by default, all objects in all CONTROL-O / COSMOS Object databases are displayed.

To activate filtering on desired criteria, specify the desired values or masks for these criteria. Mask characters (* and ?) can be specified in any of the selection criteria fields (except for EXCEPTION Y / N). Each ? represents any one character; each * represents any number of characters.

Multiple criteria are linked by an implied AND logic (meaning, an object must satisfy all specified criteria to be displayed).

All selection criteria fields (except for EXCEPTION Y / N) allow up to eight characters.

The following table contains descriptions of all fields in the window.

Table 4 Object Status Screen - Selection Criteria (part 1 of 2)

Criteria	Description
OBJECT	Object name or mask.
CURRENT	Current status or mask.
DESIRED	Desired status or mask.
CLASS	Object class or mask.

Table 4 Object Status Screen - Selection Criteria (part 2 of 2)

Criteria	Description
MODE	CONTROL-O / COSMOS mode of the object. Display only objects with the specified CONTROL-O / COSMOS mode (or with a mode that matches the specified mask).
DATABASE	Object database name or mask.
GROUP	Group name or mask.
APPL	Application name or mask.
SYSTEM NAME	Name of the Sysplex system, or mask. This name is used to select the system from which the CONTROL-O / COSMOS working databases are displayed. Default value: CURRENT (this value indicates the system on which the user is currently logged-on). This field is valid only when displaying TYPE 1 XAE databases used to store CONTROL-O / COSMOS working databases.
EXCEPTION Y / N	Indicates whether to display all objects, or only objects whose current and desired statuses do not match. Valid values are: ■ Y (Yes) – Display only objects whose current and desired statuses do not match. ■ N (No) – Display all objects (regardless of whether or not current and desired statuses match). Default.

Change selection criteria if desired. Press **Enter** to close the entry panel. The Object Status screen is displayed.

CONTROL-O/COSMOS Object Status Screen

The Object Status screen displays information about CONTROL-O / COSMOS-controlled objects.

Use the scrolling PFKeys to scroll the Object Status screen forward and backward.

Table 2-3 contains predefined display types that are available for the Object Status screen:

Table 5 Object Status Screen - Display Type

Display Type	Description
D	Default display type
S	System Name display
A	All Fields display type
O	Operator display type

Changing Display Types

While in the Object Status screen, the display type can be changed using the DISPLAY command. Format of the command is:

```
DISPLAY x
```

where *x* is the identifying letter for the desired type.

DISPLAY may be abbreviated to DI.

NOTE

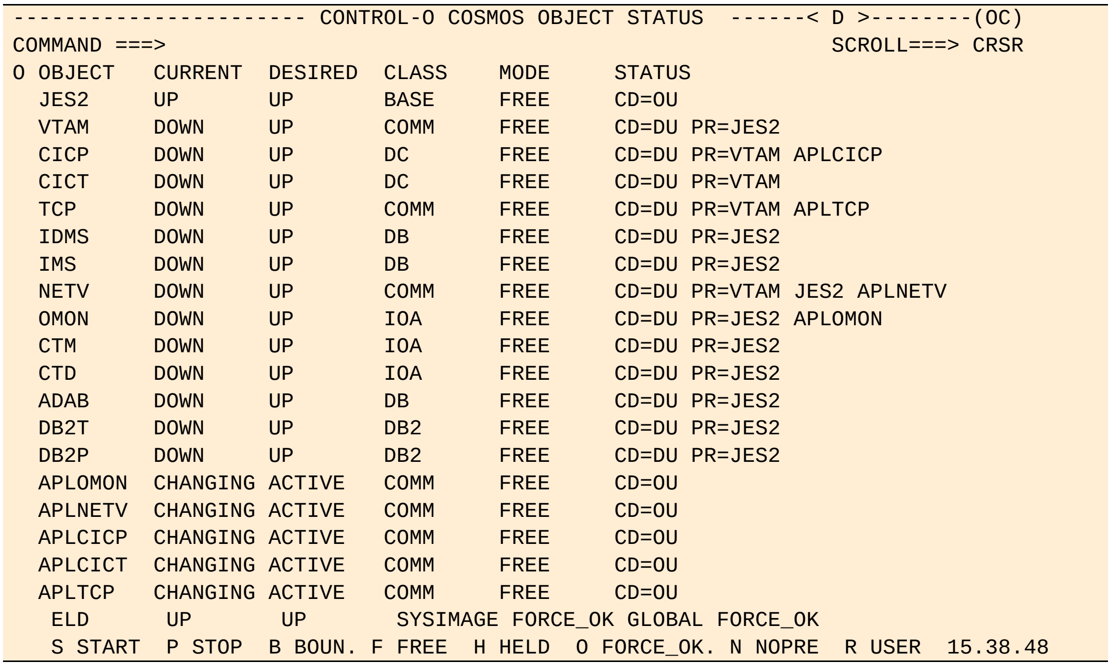
For a list of display types, enter **DISPLAY ?** to show the Display Options window. To select a display type in the window, specify S in the Option field next to the ID. To exit the window without selecting a display type, press the END key (PF3).

Example

DI A displays the All Fields display type.

Display Type D (Default)

Figure 3 Object Status Screen – Display Type D (Default)



```
----------------------- CONTROL-O COSMOS OBJECT STATUS  ------< D >--------(OC)
COMMAND ===>                                                    SCROLL===> CRSR
O OBJECT   CURRENT  DESIRED  CLASS    MODE     STATUS
  JES2     UP       UP       BASE     FREE     CD=OU
  VTAM     DOWN     UP       COMM     FREE     CD=DU PR=JES2
  CICP     DOWN     UP       DC       FREE     CD=DU PR=VTAM APLCICP
  CICT     DOWN     UP       DC       FREE     CD=DU PR=VTAM
  TCP      DOWN     UP       COMM     FREE     CD=DU PR=VTAM APLTCP
  IDMS     DOWN     UP       DB       FREE     CD=DU PR=JES2
  IMS      DOWN     UP       DB       FREE     CD=DU PR=JES2
  NETV     DOWN     UP       COMM     FREE     CD=DU PR=VTAM JES2 APLNETV
  OMON     DOWN     UP       IOA      FREE     CD=DU PR=JES2 APLOMON
  CTM      DOWN     UP       IOA      FREE     CD=DU PR=JES2
  CTD      DOWN     UP       IOA      FREE     CD=DU PR=JES2
  ADAB     DOWN     UP       DB       FREE     CD=DU PR=JES2
  DB2T     DOWN     UP       DB2      FREE     CD=DU PR=JES2
  DB2P     DOWN     UP       DB2      FREE     CD=DU PR=JES2
  APLOMON  CHANGING ACTIVE   COMM     FREE     CD=OU
  APLNETV  CHANGING ACTIVE   COMM     FREE     CD=OU
  APLCICP  CHANGING ACTIVE   COMM     FREE     CD=OU
  APLCICT  CHANGING ACTIVE   COMM     FREE     CD=OU
  APLTCP   CHANGING ACTIVE   COMM     FREE     CD=OU
   ELD      UP       UP       SYSIMAGE FORCE_OK GLOBAL FORCE_OK
   S START  P STOP  B BOUN. F FREE  H HELD  O FORCE_OK. N NOPRE  R USER  15.38.48
```

Fields in the Default (D) Display Type

The following information is displayed:

Table 6 Fields in the Default (D) Display Type (part 1 of 2)

Field	Description
<D>	Indicates the Default display type.
O (Option)	Option field. Allows entry of line options to be applied to a specific object.
OBJECT	Name of the object.
CURRENT	Current status of the object.
DESIRED	Desired status of the object
CLASS	Object class. This field can be used to create logical groups of objects, for easier management by method rules, described in Chapter 3, "Component Details."
MODE	CONTROL-O / COSMOS mode of the objects. Indicates how CONTROL-O / COSMOS is currently managing the object. Valid values are: ■ FREE – Normal operation mode. ■ NOPRE – Prerequisites are not checked before changing the status of the object. ■ FORCE_OK – Desired status of the object is modified to match the current status. If the current status is changed, the desired status is modified again so that it continues to match the current status. ■ HELD – The object is kept in the current status. No changes are made to the object status (regardless of whether or not the current status and desired status of the object match).
	Note: If a mode other than FREE is specified for a database or for all of CONTROL-O / COSMOS, that mode overrides the mode specified for specific objects. For more information, see "Database Status Screen" on page 40.

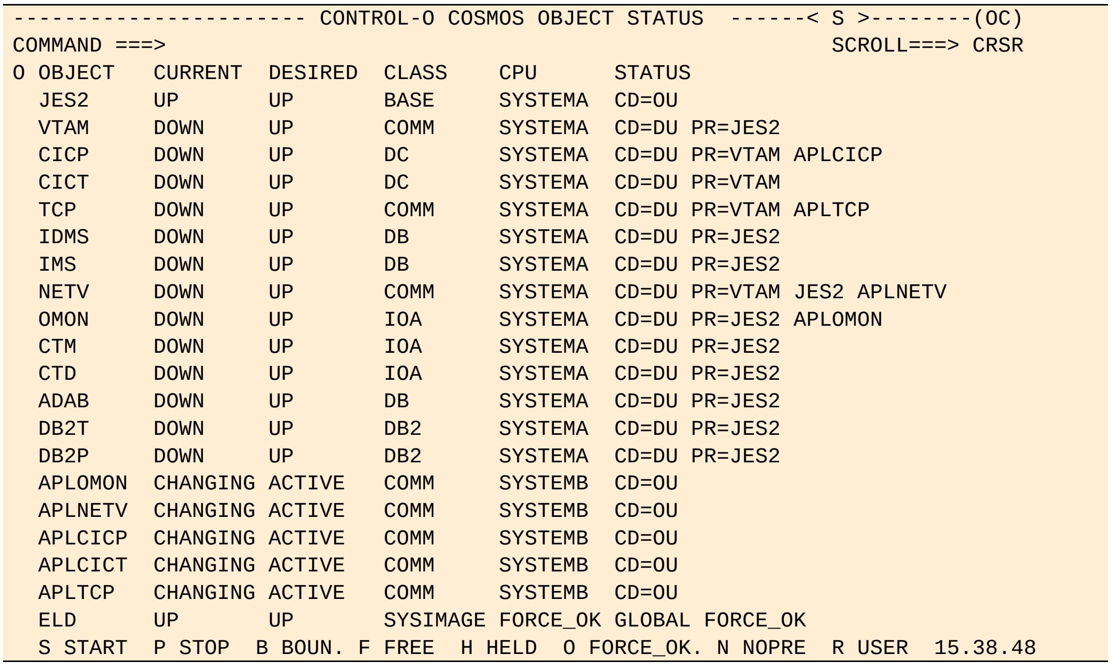
Table 6 Fields in the Default (D) Display Type (part 2 of 2)

Field	Description
STATUS	Status of the object as determined by CONTROL-O / COSMOS the last time CONTROL-O / COSMOS scanned the Object database. Valid values are: ■ STEADY UP – The current and desired statuses of the object are UP (or an equivalent status). ■ STEADY DOWN – The current and desired statuses of the object are DOWN (or an equivalent status). ■ GLOBAL HELD – HELD mode is specified either for the Object database containing the object or for the entire CONTROL-O / COSMOS facility. ■ GLOBAL FORCE_OK – FORCE_OK mode is specified either for the Object database containing the object or for the entire CONTROL-O / COSMOS facility. ■ GLOBAL NOPRE – NOPRE is specified either for the Object database containing the object or for the entire CONTROL-O / COSMOS facility.
CD=*xy* PR=*pr1 pr2 pr3* ...	The current and desired statuses of the object do not match where: ■ *xy* is an abbreviated form of the current and desired statuses. (For more information, see the CD field explanation in "Display Type A (All)" on page 34.) ■ *pr1*, *pr2*, and so on, are prerequisites that must be fulfilled before the difference between the current and desired statuses can be resolved.
	Note: If the current or desired status of the object was changed more recently than the last time CONTROL-O / COSMOS scanned the Working Object database, the value in the STATUS field may not reflect the current object status. (The STATUS field is updated the next time CONTROL-O / COSMOS scans the Working Object database.

Display Type S (System)

This display is identical to the Display Type D, except that it replaces the MODE column with CPU.

Figure 4 Object Status Screen – Display Type S (System)



```
----------------------- CONTROL-O COSMOS OBJECT STATUS  ------< S >--------(OC)
COMMAND ===>                                                    SCROLL===> CRSR
O OBJECT   CURRENT  DESIRED  CLASS    CPU      STATUS
  JES2     UP       UP       BASE     SYSTEMA  CD=OU
  VTAM     DOWN     UP       COMM     SYSTEMA  CD=DU PR=JES2
  CICP     DOWN     UP       DC       SYSTEMA  CD=DU PR=VTAM APLCICP
  CICT     DOWN     UP       DC       SYSTEMA  CD=DU PR=VTAM
  TCP      DOWN     UP       COMM     SYSTEMA  CD=DU PR=VTAM APLTCP
  IDMS     DOWN     UP       DB       SYSTEMA  CD=DU PR=JES2
  IMS      DOWN     UP       DB       SYSTEMA  CD=DU PR=JES2
  NETV     DOWN     UP       COMM     SYSTEMA  CD=DU PR=VTAM JES2 APLNETV
  OMON     DOWN     UP       IOA      SYSTEMA  CD=DU PR=JES2 APLOMON
  CTM      DOWN     UP       IOA      SYSTEMA  CD=DU PR=JES2
  CTD      DOWN     UP       IOA      SYSTEMA  CD=DU PR=JES2
  ADAB     DOWN     UP       DB       SYSTEMA  CD=DU PR=JES2
  DB2T     DOWN     UP       DB2      SYSTEMA  CD=DU PR=JES2
  DB2P     DOWN     UP       DB2      SYSTEMA  CD=DU PR=JES2
  APLOMON  CHANGING ACTIVE   COMM     SYSTEMB  CD=OU
  APLNETV  CHANGING ACTIVE   COMM     SYSTEMB  CD=OU
  APLCICP  CHANGING ACTIVE   COMM     SYSTEMB  CD=OU
  APLCICT  CHANGING ACTIVE   COMM     SYSTEMB  CD=OU
  APLTCP   CHANGING ACTIVE   COMM     SYSTEMB  CD=OU
  ELD      UP       UP       SYSIMAGE FORCE_OK GLOBAL FORCE_OK
  S START  P STOP  B BOUN. F FREE  H HELD  O FORCE_OK. N NOPRE  R USER  15.38.48
```

Fields in the Default (S) Display Type

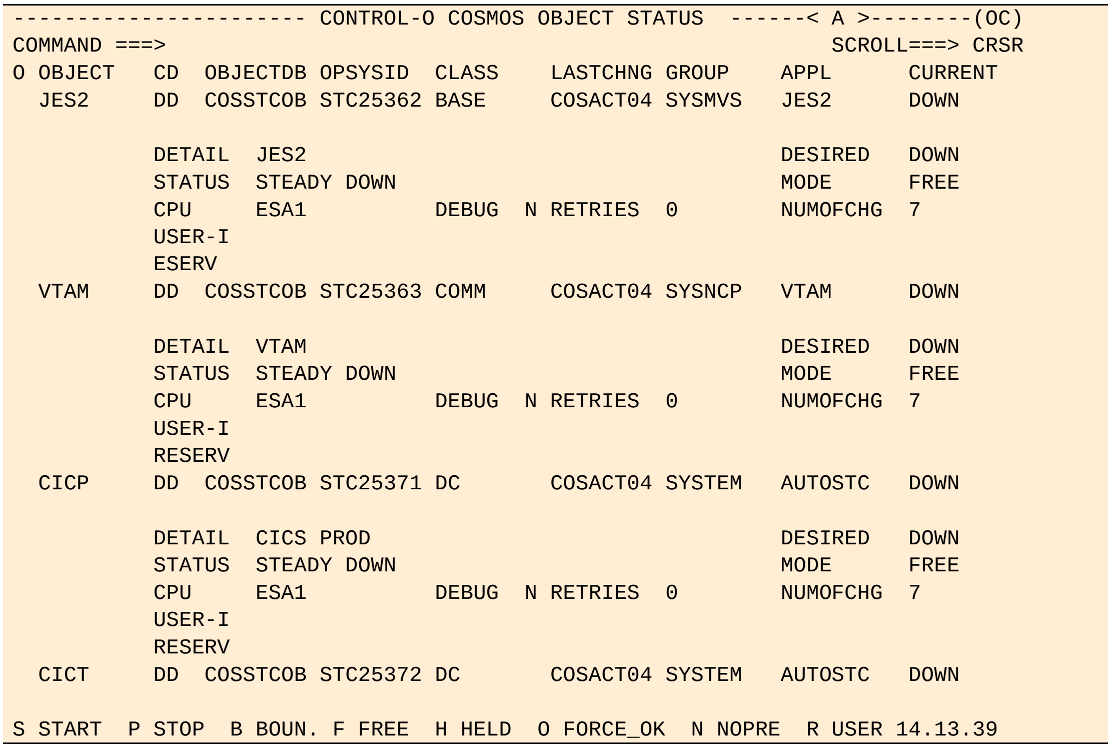
This display type contains all of the information on the Display Type D. However, in place of the MODE column, the following field and information is displayed:

CPU – Name of the system to which the managed object belongs.

Display Type A (All)

Below is an example of the All Fields display type of the Object Status screen.

Figure 5 Object Status Screen – Display Type A (All)



```
----------------------- CONTROL-O COSMOS OBJECT STATUS  ------< A >--------(OC)
COMMAND ===>                                                    SCROLL===> CRSR
O OBJECT   CD  OBJECTDB OPSYSID  CLASS    LASTCHNG GROUP    APPL      CURRENT
  JES2     DD  COSSTCOB STC25362 BASE     COSACT04 SYSMVS   JES2      DOWN

           DETAIL  JES2                                     DESIRED   DOWN
           STATUS  STEADY DOWN                              MODE      FREE
           CPU     ESA1          DEBUG  N RETRIES  0        NUMOFCHG  7
           USER-I
           ESERV
  VTAM     DD  COSSTCOB STC25363 COMM     COSACT04 SYSNCP   VTAM      DOWN

           DETAIL  VTAM                                     DESIRED   DOWN
           STATUS  STEADY DOWN                              MODE      FREE
           CPU     ESA1          DEBUG  N RETRIES  0        NUMOFCHG  7
           USER-I
           RESERV
  CICP     DD  COSSTCOB STC25371 DC       COSACT04 SYSTEM   AUTOSTC   DOWN

           DETAIL  CICS PROD                                DESIRED   DOWN
           STATUS  STEADY DOWN                              MODE      FREE
           CPU     ESA1          DEBUG  N RETRIES  0        NUMOFCHG  7
           USER-I
           RESERV
  CICT     DD  COSSTCOB STC25372 DC       COSACT04 SYSTEM   AUTOSTC   DOWN

S START  P STOP  B BOUN. F FREE  H HELD  O FORCE_OK  N NOPRE  R USER 14.13.39
```

Fields in the All Fields (A) Display Type

The following information is displayed:

Table 7 Fields in the All Fields (A) Display Type (part 1 of 3)

Field	Description
<A>	Indicates the All Fields display type.
O (Option)	Allows entry of line options to be applied to a specific object.
OBJECT	Name of the object.

Table 7 Fields in the All Fields (A) Display Type (part 2 of 3)

Field	Description
CD	Abbreviated combination of the current and desired statuses of the objects. The value appearing in this field is in format: *xy* where ■ *x*: One-character value indicating the current status. ■ *y*: One-character value indicating the desired status. Valid values for *x* and *y*: ■ D: Down status (or a specified synonym). ■ U: Up status (or a specified synonym). ■ ?: Unknown status (or a specified synonym). ■ O: Other status (for example, CHANGING, or STARTING).
OBJECTDB	Name of the Object database in which the object is defined.
OPSYSID	ID assigned by the operating system to the current run of the object (for example, job ID of the started task).
CLASS	Object class. This field can be used to create logical groups of objects, for easier management by method rules, described in Chapter 3, “Component Details.”
LASTCHNG	Name of the CONTROL-O/COSMOS rule that last changed the object’s status or mode
GROUP	User-defined group name for the object.
APPL	User-defined name for the application of the object. This field can be used as a common descriptive name for object groups (meaning, for one or more different groups).
CURRENT	Current status of the object.
DETAIL	Free text description of the object.
DESIRED	Desired status of the object.
STATUS	Status of the object as determined by CONTROL-O/COSMOS’s analysis of the object the last time CONTROL-O/COSMOS scanned the Object database.

Table 7 Fields in the All Fields (A) Display Type (part 3 of 3)

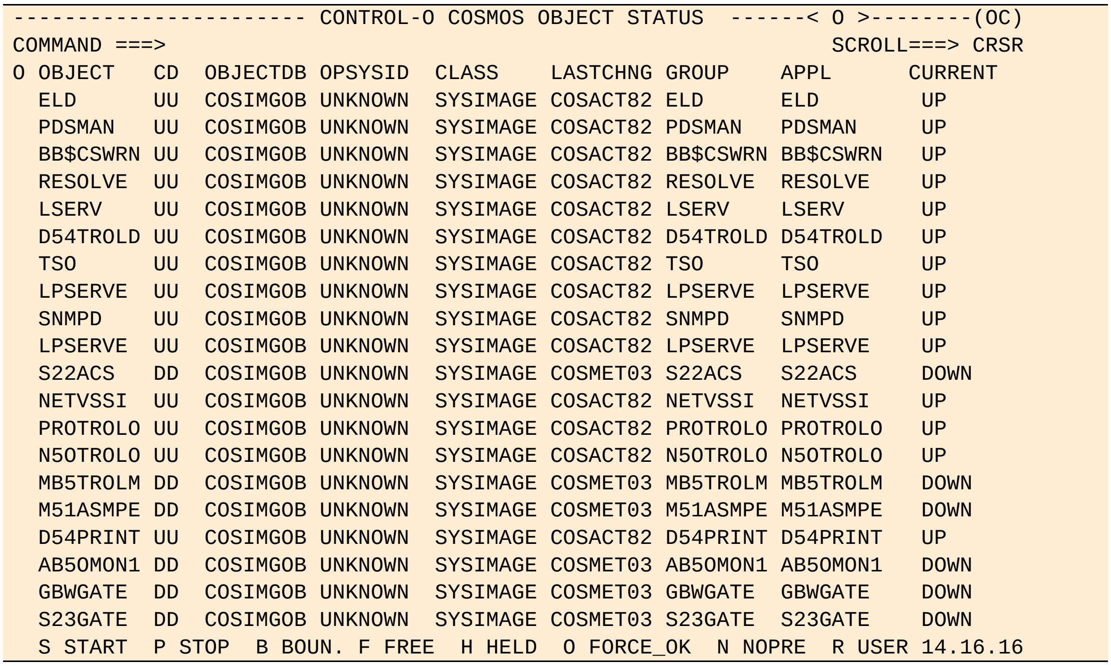
Field	Description
MODE	CONTROL-O / COSMOS mode of the objects. Indicates how CONTROL-O / COSMOS is currently managing the object. valid values: ■ FREE – Normal operation mode. ■ NOPRE – Prerequisites are not checked before changing the status of the object. ■ FORCE_OK – Desired status of the object is modified to match the current status. If the current status is changed, the desired status is modified again so that it continues to match the current status. ■ HELD – The object is kept in the current status. No changes are made to the object status (regardless of whether or not the current status and desired status of the object match). ■ If a mode other than FREE is specified for a database or for all of CONTROL-O / COSMOS, that mode overrides the mode specified for specific objects. For more information, see “Database Status Screen” on page 40.
CPU	ID of the CPU on which the object is running.
DEBUG	Whether or not the object is running in DEBUG mode. If the object is running in debug mode, a shout message is issued each time a change is made to the object’s status.
RETRIES	Number of times a CONTROL-O / COSMOS action has been attempted unsuccessfully (meaning, the number of unsuccessful attempts made to change the current status of the object since the last time its status was changed).
NUMOFCHG	Number of times the status of the object changed since CONTROL-O / COSMOS was activated.
USER-I	User supplied data or information, if desired (taken from column USERDAT1 in the database. The contents of this field do not affect CONTROL-O / COSMOS.
RESERV	Reserved for future use (in column COSRSRV1 in the database).

Display Type O (Operator)

Below is an example of the Operator display type of the Object Status screen.

Like the Default display type, the Operator display type describes each object on one line of the screen. However, different fields (meaning, fields more relevant to the operator) are displayed.

Figure 6 Object Status Screen – Display Type O (Operator)



```
----------------------- CONTROL-O COSMOS OBJECT STATUS  ------< O >--------(OC)
COMMAND ===>                                                    SCROLL===> CRSR
O OBJECT   CD  OBJECTDB OPSYSID  CLASS    LASTCHNG GROUP    APPL      CURRENT
  ELD      UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 ELD      ELD        UP
  PDSMAN   UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 PDSMAN   PDSMAN     UP
  BB$CSWRN UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 BB$CSWRN BB$CSWRN   UP
  RESOLVE  UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 RESOLVE  RESOLVE    UP
  LSERV    UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 LSERV    LSERV      UP
  D54TROLD UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 D54TROLD D54TROLD   UP
  TSO      UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 TSO      TSO        UP
  LPSERVE  UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 LPSERVE  LPSERVE    UP
  SNMPD    UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 SNMPD    SNMPD      UP
  LPSERVE  UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 LPSERVE  LPSERVE    UP
  S22ACS   DD  COSIMGOB UNKNOWN  SYSIMAGE COSMET03 S22ACS   S22ACS     DOWN
  NETVSSI  UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 NETVSSI  NETVSSI    UP
  PROTROLO UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 PROTROLO PROTROLO   UP
  N50TROLO UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 N50TROLO N50TROLO   UP
  MB5TROLM DD  COSIMGOB UNKNOWN  SYSIMAGE COSMET03 MB5TROLM MB5TROLM   DOWN
  M51ASMPE DD  COSIMGOB UNKNOWN  SYSIMAGE COSMET03 M51ASMPE M51ASMPE   DOWN
  D54PRINT UU  COSIMGOB UNKNOWN  SYSIMAGE COSACT82 D54PRINT D54PRINT   UP
  AB50MON1 DD  COSIMGOB UNKNOWN  SYSIMAGE COSMET03 AB50MON1 AB50MON1   DOWN
  GBWGATE  DD  COSIMGOB UNKNOWN  SYSIMAGE COSMET03 GBWGATE  GBWGATE    DOWN
  S23GATE  DD  COSIMGOB UNKNOWN  SYSIMAGE COSMET03 S23GATE  S23GATE    DOWN
  S START  P STOP  B BOUN. F FREE  H HELD  O FORCE_OK  N NOPRE  R USER 14.16.16
```

Fields in the (O) Display Type

The following information is displayed:

Table 8 Fields in Display Type O (part 1 of 2)

Field	Description
<O>	Indicates the Operator display type.
O (Option)	Allows entry of line options to be applied to a specific object.
OBJECT	Name of the object.
CD	Abbreviated combination of the current and desired statuses of the object. For more information, see the description of this field in "Display Type A (All)" on page 34.
OBJECTDB	Name of the Object database in which the object is defined.
OPSYSID	ID assigned by the operating system to the current run of the object (for example, job ID of the started task).

Table 8 Fields in Display Type O (part 2 of 2)

Field	Description
CLASS	Object class. This field can be used to create logical groups of objects. CONTROL-O / COSMOS methods can be created for management of specific objects, or all objects in a specific class. For more information about CONTROL-O / COSMOS methods, see Chapter 3, “Component Details.”
LASTCHNG	Name of the CONTROL-O / COSMOS rule that last changed the object’s status or mode.
GROUP	User-defined group name for the object.
APPL	User-defined name for the application of the object. This field can be used as a common descriptive name for object groups (meaning, for one or more different groups).
CURRENT	Current status of the object.

Options of the Object Status Screen

The following valid options, listed at the bottom of the screen, can be applied to any row in the Object Status screen:

Table 9 Options of the Object Status Screen

Code	Option	Description
S	(START)	Set the Desired status of the object to UP.
P	(STOP)	Set the Desired status of the object to DOWN.
B	(BOUNCE)	Recycle an object by first setting the desired status to DOWN and then (after the object is brought down) resetting it to UP.
F	(FREE)	Set the CONTROL-O / COSMOS mode of the object to FREE.
H	(HELD)	Set the CONTROL-O / COSMOS mode of the object to HELD.
O	(FORCE_OK)	Set the CONTROL-O / COSMOS mode of the object to FORCE_OK.
N	(NOPRE)	Set the CONTROL-O / COSMOS mode of the object to NOPRE.
R	(USER)	Triggers a user-defined action (rule) on the object. Choosing this options triggers rule COSACT98 that is supplied with CONTROL-O / COSMOS. This rule can be customized by the user to perform an action not handled by predefined options. For more information, see Chapter 3, “Component Details.”

NOTE

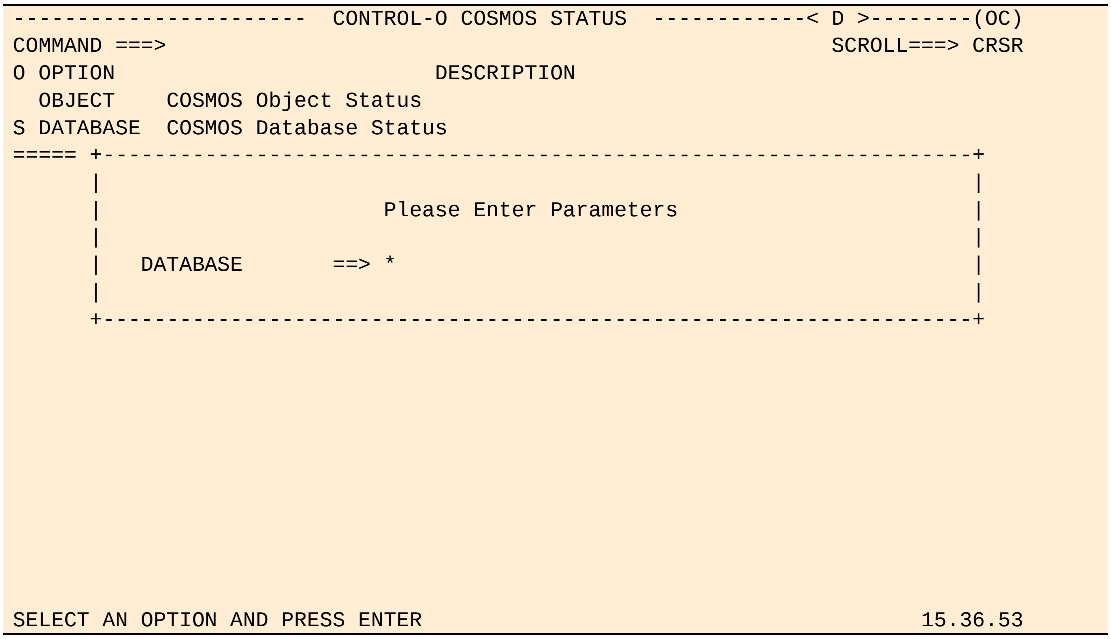
UP and DOWN statuses mentioned above refer to the values specified for UP and DOWN statuses for the relevant Object database in member COSMOLST. For more information on the various status names, see Chapter 3, “Component Details.”

CONTROL-O/COSMOS Database Status Screen – Selection Window

The Database Status screen displays information about CONTROL-O / COSMOS Object Databases. Options in the Database Status screen can be used to modify the mode of an Object database or of the entire CONTROL-O / COSMOS facility.

To display the Database Status screen, specify **S** (Select) to the left of option DATABASE in the CONTROL-O / COSMOS Main menu and press **Enter**. The following window is displayed.

Figure 7 Database Status Screen – Selection Window



```
-----------------------  CONTROL-O COSMOS STATUS  ------------< D >--------(OC)
COMMAND ===>                                                    SCROLL===> CRSR
O OPTION                         DESCRIPTION
  OBJECT    COSMOS Object Status
S DATABASE  COSMOS Database Status
===== +--------------------------------------------------------------------+
      |                                                                    |
      |                     Please Enter Parameters                        |
      |                                                                    |
      |   DATABASE       ==> *                                             |
      |                                                                    |
      +--------------------------------------------------------------------+











SELECT AN OPTION AND PRESS ENTER                                       15.36.53
```

Specify the name (or mask) of the Object databases to be displayed and press **ENTER**. The Database Status screen is displayed.

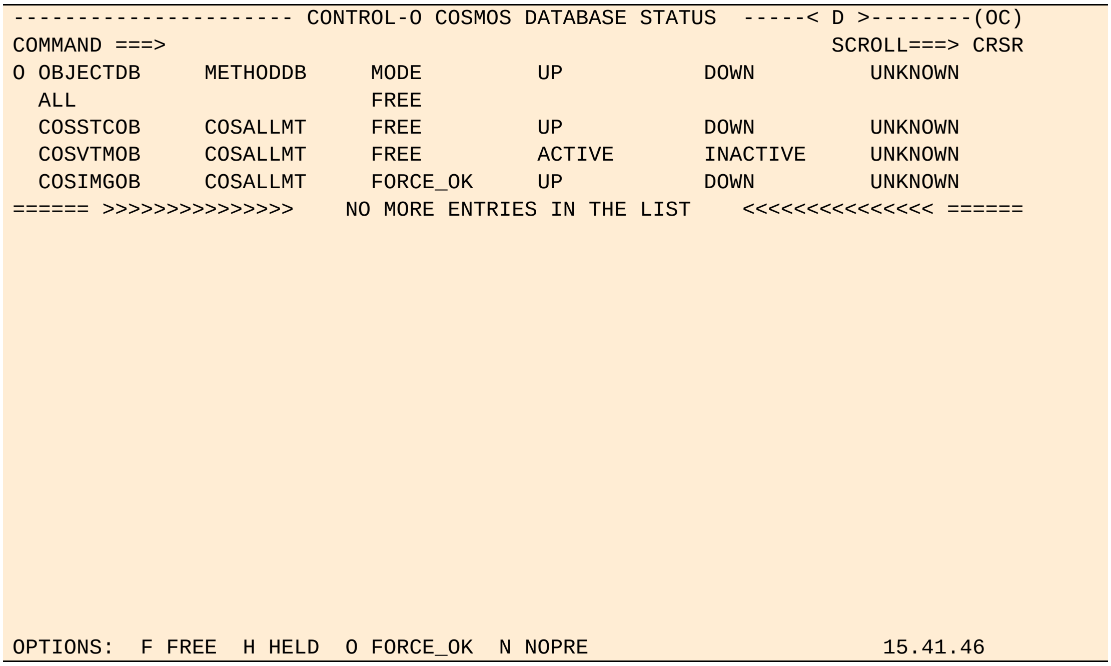
Database Status Screen

The Database Status screen contains basic information about CONTROL-O / COSMOS Object databases. Options in this screen are similar to the options that can be specified in the Object Status screen (described above). However, in this screen, the options affect the mode of a specified database.

In addition, the first record in this screen (ALL) can be used to specify options that affect Global CONTROL-O / COSMOS mode.

Use the scrolling PFKeys to scroll the Database Status screen forward and backward.

Figure 8 Database Status Screen



```
---------------------- CONTROL-O COSMOS DATABASE STATUS  -----< D >--------(OC)
COMMAND ===>                                                     SCROLL===> CRSR
O OBJECTDB     METHODDB     MODE         UP           DOWN         UNKNOWN
  ALL                       FREE
  COSSTCOB     COSALLMT     FREE         UP           DOWN         UNKNOWN
  COSVTMOB     COSALLMT     FREE         ACTIVE       INACTIVE     UNKNOWN
  COSIMGOB     COSALLMT     FORCE_OK     UP           DOWN         UNKNOWN
====== >>>>>>>>>>>>>>>    NO MORE ENTRIES IN THE LIST    <<<<<<<<<<<<<<< ======






























OPTIONS:  F FREE  H HELD  O FORCE_OK  N NOPRE                       15.41.46
```

The information included in the Database Status screen is built from member COSMOLST in the CONTROL-O PARM library. For information about the format of this member and how to modify the information, see Chapter 3, “Component Details.”

The following fields are displayed in the Database Status screen:

Table 10 Fields of the Database Status Screen (part 1 of 2)

Field	Description
O (Option)	Option field. Allows entry of line options to be applied to a specific object.
OBJECTDB	Name of the Object database.

Table 10 Fields of the Database Status Screen (part 2 of 2)

Field	Description
METHODDB	Name of the Method database used to determine actions for the Object database.
MODE	CONTROL-O / COSMOS mode for the database. Valid values are: ■ FREE – Normal operation mode. ■ HELD – Objects in the database are kept in the current status. No changes are made to the status of objects in the database (regardless of whether or not the current status and desired status of the objects match) ■ FORCE_OK – Desired status of objects in the database are modified to match the current status. If the current status of an object is changed, its desired status is modified again so that it continues to match the current status ■ NOPRE – Prerequisites are not checked before changing the status of objects in the database.
UP	Name of the UP status for objects in the database (for example, UP, ONLINE, or ACTIVE).
DOWN	Name of the DOWN status for objects in the database (for example, DOWN, OFFLINE, or INACTIVE).
UNKNOWN	Name of the UNKNOWN status for objects in the database (for example, UNKNOWN, or NOTFOUND).

Options of the Database Status Screen

The following valid options, listed at the bottom of the screen, can be applied to any row in the Database Status screen:

Table 11 Database Status Screen Options

Code	Option	Description
F	(FREE)	Sets the CONTROL-O / COSMOS mode of the database to FREE.
H	(HELD)	Sets the CONTROL-O / COSMOS mode of the database to HELD.
O	(FORCE_OK)	Sets the CONTROL-O / COSMOS mode of the database to FORCE_OK.
N	(NOPRE)	Sets the CONTROL-O / COSMOS mode of the database to NOPRE.

Specifying an Option for the Entire CONTROL-O/COSMOS Facility

The first row in the Database Status screen can be used to specify the mode in which all CONTROL-O / COSMOS-controlled objects are handled. Specify a valid option to the left of the first record in the screen (with ALL in the OBJDB column) to apply an option to all CONTROL-O / COSMOS-controlled objects.

NOTE

When a mode is specified for all of CONTROL-O / COSMOS, the mode is not changed in each record in the Object databases, but the objects are handled as if the specified mode was indicated for them.

If mode FREE is specified for the first record in the Database Status screen, the mode of each Object database determines the handling of objects.

If mode FREE is specified for the first record in the Database Status screen, and mode FREE is specified for an Object database, objects defined in the database are handled according to the mode specified in each record of that Object database (meaning, object-specific mode).

Chapter 3

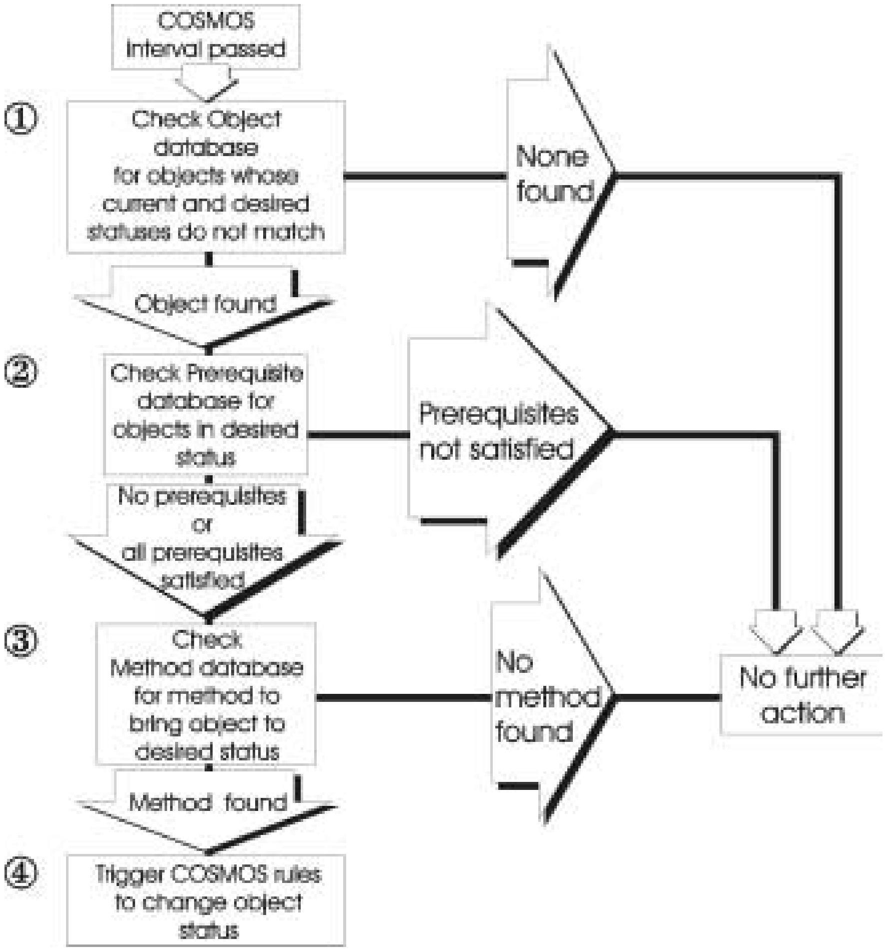
Component Details

This chapter includes the following topics:

CONTROL-O/COSMOS Workflow

Normally CONTROL-O / COSMOS brings objects to a predefined desired status during IPL. However, the production environment is dynamic, and the desired status of objects can change. The following workflow is therefore used to manage CONTROL-O / COSMOS-controlled objects:

Figure 9 CONTROL-O/COSMOS Workflow


COSMOS interval passed
①
Check Object database for objects whose current and desired statuses do not match
None found
Object found
②
Check Prerequisite database for objects in desired status
Prerequisites not satisfied
No prerequisites or all prerequisites satisfied
③
Check Method database for method to bring object to desired status
No method found
No further action
Method found
④
Trigger COSMOS rules to change object status

1. CONTROL-O / COSMOS periodically scans the Object database.

2. If an object has a current status of UP and a desired status of DOWN, or vice versa, CONTROL-O / COSMOS scans the Prerequisite database to determine the prerequisites for a change to the desired status, and then checks the relevant Object databases to determine if the prerequisites are satisfied.

   If prerequisites for an object are not satisfied, no action is performed on that object.

3. CONTROL-O / COSMOS scans the Method database to determine the method for changing the status of the object to its desired status.

If no method is found that matches an object in its desired status, no action is performed on the object.

4. If an appropriate method is found, CONTROL-O / COSMOS triggers the CONTROL-O / COSMOS rule that performs the method.

User requests (for example, CONTROL-O / COSMOS commands or options in a CONTROL-O / COSMOS online screen) can also be used to trigger rules that modify the contents of a CONTROL-O / COSMOS Object database.

Example

Assume that object CICSTEST has a current status of DOWN and its desired status is DOWN. No further action (regarding CICSTEST) is necessary. However, if the operator changes the desired status of CICSTEST to UP, the following events occur:

1. The next time CONTROL-O / COSMOS scans the Object database, it notes that CICSTEST's current status (DOWN) does not match its desired status (UP).

2. CONTROL-O / COSMOS checks the prerequisites for CICSTEST with a status of UP (for example, VTAM is up). If prerequisites are not satisfied, no action is performed on object CICSTEST.

3. If the prerequisites are satisfied, CONTROL-O / COSMOS searches the Method database for the method to change CICSTEST's status to UP. Assume that the Method rule found is named CICSUP.

4. A CONTROL-O / COSMOS rule changes the status of CICSTEST to CHANGING.

5. CONTROL-O / COSMOS triggers rule CICSUP to bring CICSTEST up.

6. When the CICSTEST address space starts, the following message is issued:

   ```
   IEF403I CICSTEST - STARTED - TIME=HH.MM.SS
   ```

   A CONTROL-O / COSMOS rule detects this message, and changes the current status of CICSTEST to STARTING.

7. The CICSTEST address space issues the following message:

   ```
   DFHSI1517 CICSTEST CONTROL IS BEING GIVEN TO CICS
   ```

8. A CONTROL-O / COSMOS rule detects message DFHSI1517 and sets the current status of CICSTEST to UP.

Object CICSTEST remains in status UP, until the desired status is changed (for example, using an operator command, or using a CONTROL-O / COSMOS online screen).

NOTE

When an object has a status of CHANGING or STARTING, CONTROL-O / COSMOS detects that the desired and current statuses do not match. However, no action is taken since no methods are defined for a current status of CHANGING or STARTING

CONTROL-O/COSMOS Modes

A CONTROL-O / COSMOS mode is specified for each CONTROL-O / COSMOS-controlled object in the Object databases. The CONTROL-O / COSMOS mode determines if the object is handled in a special way (for example, if prerequisites are ignored). For example, a CONTROL-O / COSMOS mode can indicate not to modify an object or not to check prerequisites before an object is modified.

A CONTROL-O / COSMOS mode can be modified using an option in a CONTROL-O / COSMOS screen, or using a CONTROL-O / COSMOS command such as COSDB or COSCMD, described on pages page 80 and page 81, respectively.

The following modes are available:

Table 12 CONTROL-O/COSMOS Modes (part 1 of 2)

Mode	Description
FREE	Normal CONTROL-O / COSMOS operations are performed on the object.
HELD	Rules are not applied by CONTROL-O / COSMOS to the object. This mode can be used if you need to temporarily disable automation rules that update an object’s status.
	Note: When a held object is assigned a different mode (for example, FREE), its current status is not known and is therefore set to UNKNOWN. The next time CONTROL-O / COSMOS scans the Object database, it generally triggers a CONTROL-O / COSMOS rule that determines the current status of the object.

Table 12 CONTROL-O/COSMOS Modes (part 2 of 2)

Mode	Description
FORCE_OK	CONTROL-O / COSMOS forces the object's desired status to take the value of its current status. If the current status is changed, the desired status is modified so that it again matches the current status. When CONTROL-O / COSMOS is started following (rather than during) an IPL, mode FORCE_OK is used to indicate that CONTROL-O / COSMOS uses the current status as a starting point. For more information, see "COSINI Command – Starting CONTROL-O / COSMOS" on page 77.
NOPRE	CONTROL-O / COSMOS does not check prerequisites before looking for a method matching the object's current and desired status. This mode is useful for temporarily overriding dependencies among computing objects.

A CONTROL-O / COSMOS mode is also specified for each Object database and for the entire CONTROL-O / COSMOS facility.

The mode specified for the CONTROL-O / COSMOS facility has a higher priority than the mode specified for each Object database.

- If mode HELD, FORCE_OK or NOPRE is specified for the CONTROL-O / COSMOS facility, this mode overrides the mode specified for each Object database.
- If mode FREE is specified for the CONTROL-O / COSMOS facility, the mode specified for each Object database determines the handling of CONTROL-O / COSMOS-controlled objects.
  The mode specified for each Object database has a higher priority than the mode specified for each CONTROL-O / COSMOS-controlled object.
- If mode HELD, FORCE_OK or NOPRE is specified for the Object database, this mode overrides the mode specified for each object.
- If mode FREE is specified for CONTROL-O / COSMOS, and mode FREE is specified for an Object database, objects defined in the database are handled according to the mode specified in each row of that Object database (that is, object-specific mode).

When a mode is specified for all of CONTROL-O / COSMOS or for an Object database, the mode is not changed in each row in the Object databases.

CONTROL-O/COSMOS Databases

The following types of databases must be created:

- At least one Object database
- At least one Method database
- The Prerequisite database

Member COSMOLST in the CONTROL-O PARM library contains definitions of all databases used by CONTROL-O/COSMOS. When a new database is created, a row describing it must be added to member COSMOLST, described in "CONTROL-O/COSMOS Initialization Rules" on page 66.

Detailed descriptions of the CONTROL-O/COSMOS databases, including their creation and maintenance, are provided below.

Creating and Maintaining Object Databases

CONTROL-O/COSMOS Object databases contain information about the objects (started tasks, terminals, and so on) managed by CONTROL-O/COSMOS.

Object databases are normally created in pairs. Each pair consists of the following:

- Source Object database

  Contains all object names and pertinent information as a permanent database.

- Working Object database

  Working copy of the Source Object database used to manage objects during CONTROL-O/COSMOS operations. This copy is modified by CONTROL-O/COSMOS at regular intervals to reflect changes in the production environment.

Working Object databases are required and are used by CONTROL-O/COSMOS. Source Object databases can be used, but are not required.

CONTROL-O/COSMOS determines what actions are performed, by scanning the Working Object databases. All information in the Working Object databases is copied from the Source Object database each time CONTROL-O/COSMOS is activated. You can use other means to create the Working Object database. For example, the Working Object database can be created dynamically.

Creating rows with a rule can create a Working Object database. For example, an operator command can be issued to display objects and a command response can be used to add rows to the list of objects to be managed.

An Object database, describing started task objects at your site, can be created using the SYSIMAGE facility, and can be maintained through the Variable Database Definition facility. This Object database can be modified, and other Object databases can be created, using the Variable Database facility.

Maintaining separate Source and Working Object databases provides the following advantages:

- Saves disk space.

  The Working Object database contains much information that is meaningless after CONTROL-O / COSMOS is brought down. The Source Object database contains only the information necessary for re-creation of Working Object databases.

- Enables management of objects using one Source Object database.

  The Source Object database is used to upload information for multiple Working Object databases (that is, one on each CPU) when CONTROL-O / COSMOS is started. Using one Source Object database avoids duplicate definitions in multi-CPU and Sysplex environments.

Example

VTAM normally depends on JES2 no matter which CPU of the system is involved. Therefore, if an object named JES2 is appropriately defined in the Source Object database, the definition of JES2 is copied to the Working Object database of every CPU.

Source Object Databases

When CONTROL-O / COSMOS is started, information is copied from the Source Object databases to the Working Object databases.

Addition, modification and deletion of CONTROL-O / COSMOS-controlled objects are performed on only the Source Object databases. These operations are performed using the Variable Database Definition facility (screen IV), described in Chapter 2, “Online Facilities.”

Each object is described by a row in the Source Object database. The following columns must be included in the Source Object database:

Table 13 Required Columns in the Source Object Database (part 1 of 2)

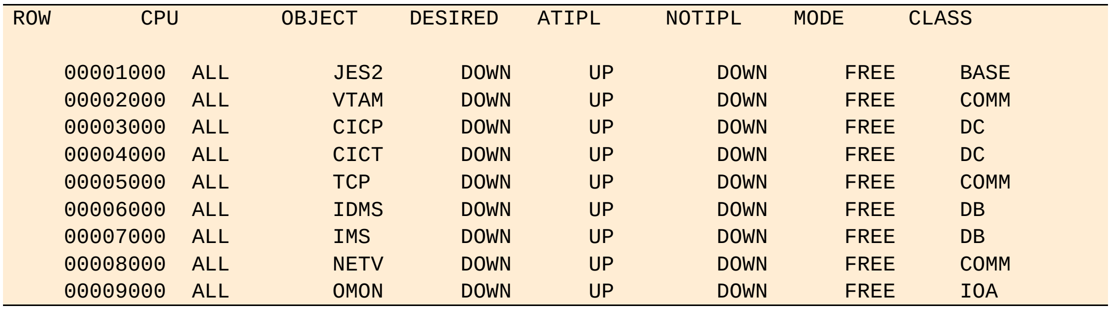
Column	Description
OBJECT	Name of the object (for example, CICSTEST, CICSPROD, JES2, LU1290).
DESIRED	Desired status of the object (for example, UP, or DOWN).
MODE	Object operating mode. For more information, see "CONTROL-O / COSMOS Rules" on page 63. valid values: ■ FREE – CONTROL-O / COSMOS performs normal operations on the object. ■ HELD – CONTROL-O / COSMOS does not perform any operations on the object. ■ FORCE_OK – CONTROL-O / COSMOS changes the desired status to match the current status. If the current status is changed, CONTROL-O / COSMOS modifies the desired status to match the new current status. ■ NOPRE – CONTROL-O / COSMOS does not check prerequisites before searching the Method database for a method to change the objects current state to its desired state.
CLASS	Class of the object (for example, CICS). This field is used as an alternative to matching the object's name when searching for a method to change the current status.
DETAIL	Free text description of the object.
DEBUG	Generate debugging information created when CONTROL-O / COSMOS handles the object. If Y is specified, the Debug facility is activated and a SHOUT message is issued each time a CONTROL-O / COSMOS-controlled object's status is changed.
GROUP	User-defined group name for the object.
APPL	User-defined application name for the application of the object. This field can be used as a common descriptive name for object groups (meaning, for one or more different groups).
CPU	ID of a CPU, or ALL. This field is used to determine which objects are copied to the Working Object databases in each CPU. If ALL is specified for an object, that object is copied to the Working Object database in each CPU.
ATIPL	Value of the desired status if CONTROL-O / COSMOS is started during an IPL.
NOTIPL	Value of the desired status if CONTROL-O / COSMOS is not started by during IPL.
USERDAT1	Optional user-supplied information. This information does not affect CONTROL-O / COSMOS. This value appears in the USER-I field in the All Fields display type of the CONTROL-O / COSMOS Objects Status screen.

Table 13 Required Columns in the Source Object Database (part 2 of 2)

Column	Description
USERDAT2	Same as USERDAT1, but not shown in the All Fields display type.
COSRSRV1	Reserved for future use. The value is shown in the RESERV field in the All Fields display type of the CONTROL-O / COSMOS Objects Status screen.
COSRSRV2	Same as COSRSRV1, but not shown in the All Fields display type.

The following screen segment contains sample content of a Source Object database (as displayed in the Variable Database screen):

Figure 10 Source Object Database Example



```
ROW       CPU        OBJECT    DESIRED   ATIPL     NOTIPL    MODE     CLASS

    00001000  ALL        JES2      DOWN      UP        DOWN      FREE     BASE
    00002000  ALL        VTAM      DOWN      UP        DOWN      FREE     COMM
    00003000  ALL        CICP      DOWN      UP        DOWN      FREE     DC
    00004000  ALL        CICT      DOWN      UP        DOWN      FREE     DC
    00005000  ALL        TCP       DOWN      UP        DOWN      FREE     COMM
    00006000  ALL        IDMS      DOWN      UP        DOWN      FREE     DB
    00007000  ALL        IMS       DOWN      UP        DOWN      FREE     DB
    00008000  ALL        NETV      DOWN      UP        DOWN      FREE     COMM
    00009000  ALL        OMON      DOWN      UP        DOWN      FREE     IOA
```

NOTE

Only seven columns of the Source Object database can be displayed at one time. Additional columns can be viewed by using the RIGHT and LEFT scrolling conventions.

Information in the Source Object database is copied to columns with the same names in the Working Object database. If a column exists in the Source Object database, and no column with the same name is found in the Working Object database, information in that column is not copied.

NOTE

Columns ATIPL, NOTIPL and DESIRED (described above) are not copied to the Working Object database. These fields are used by CONTROL-O / COSMOS initialization procedures to determine the startup value for the DESIRED column in the Working Object database. The CONTROL-O / COSMOS startup option is specified in response to a WTOR message issued by command COSINI, described on page 77.

Working Object Databases

Working Object databases are scanned periodically by CONTROL-O / COSMOS to determine if specified objects are in their desired statuses. When CONTROL-O / COSMOS is activated, rows containing object-related information are copied from the Source Object Databases to Working Object databases.

Most columns in Working Object databases contain information copied from the Source Object databases. However, certain additional columns are added for information required for ongoing CONTROL-O / COSMOS operations.

Each Working Object database must contain all columns in the Source Object database except for ATIPL and NOTIPL. In addition, the following columns must appear in the Working Object database:

Table 14 Working Object Database – Column Descriptions

Column	Description
CURRENT	Current status of the object.
DESIRED	Desired status of the object. The value inserted in this field can be copied from a specific Source Object database column (ATIPL, NOTIPL or DESIRED) or it can be set according to a global preference, depending on the option chosen when CONTROL-O / COSMOS is started.
STATUS	Text created by CONTROL-O / COSMOS with object-related information.
LASTCHNG	Name of the CONTROL-O / COSMOS rule that last changed object status or mode.
OPSYSID	ID assigned by the operating system to the current run of the object (for example, job ID of the started task).
RETRIES	Number of times a CONTROL-O / COSMOS action has been attempted unsuccessfully (meaning, the number of unsuccessful attempts made to change the current status of the object since the last time its status was changed).
NUMOFCHG	Number of times the status of the object changed since CONTROL-O / COSMOS was activated.

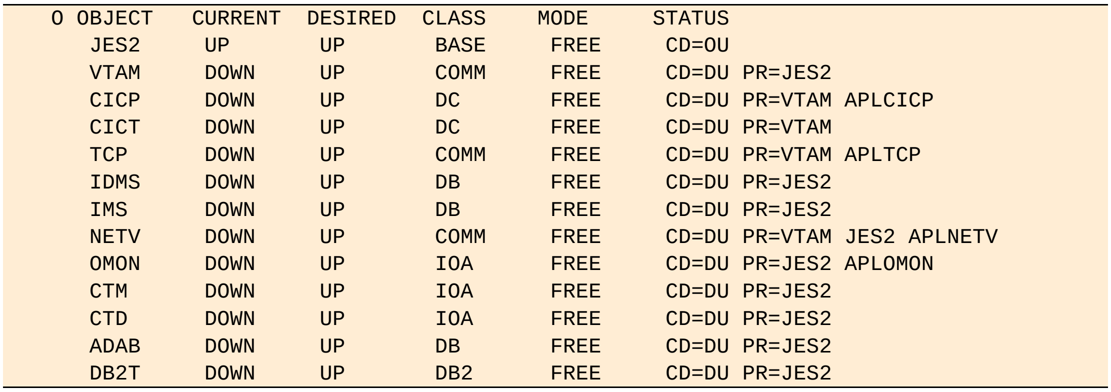
For started task objects, it is recommended that the following columns be included:

Table 15 Started Task Objects – Columns

Column	Description
REPLYID	Last reply ID.
COMMCHAR	Command character (when necessary to issue operator commands to the started task).

The following screen segment contains sample content of a Working Object database (as displayed in the Object Status screen):

Figure 11 Working Object Database Example



```
 O OBJECT    CURRENT  DESIRED  CLASS    MODE     STATUS
   JES2       UP       UP       BASE     FREE     CD=OU
   VTAM       DOWN     UP       COMM     FREE     CD=DU PR=JES2
   CICP       DOWN     UP       DC       FREE     CD=DU PR=VTAM APLCICP
   CICT       DOWN     UP       DC       FREE     CD=DU PR=VTAM
   TCP        DOWN     UP       COMM     FREE     CD=DU PR=VTAM APLTCP
   IDMS       DOWN     UP       DB       FREE     CD=DU PR=JES2
   IMS        DOWN     UP       DB       FREE     CD=DU PR=JES2
   NETV       DOWN     UP       COMM     FREE     CD=DU PR=VTAM JES2 APLNETV
   OMON       DOWN     UP       IOA      FREE     CD=DU PR=JES2 APLOMON
   CTM        DOWN     UP       IOA      FREE     CD=DU PR=JES2
   CTD        DOWN     UP       IOA      FREE     CD=DU PR=JES2
   ADAB       DOWN     UP       DB       FREE     CD=DU PR=JES2
   DB2T       DOWN     UP       DB2      FREE     CD=DU PR=JES2
```

NOTE

This sample shows the Default display type of the Object Status screen. Additional columns of the Working Object database can be viewed using other display types of this screen. For more information, see "CONTROL-O / COSMOS Object Status Screen" on page 29.

Multiple Source and Working Object Databases

Multiple Source Object databases can be defined for CONTROL-O / COSMOS objects. The number of Source Object databases appropriate for your site depends on the nature of the objects to be managed and the way in which CONTROL-O / COSMOS is to be implemented. Definition of multiple Source Object databases enables you to:

- Separate objects of different types (for example, started tasks and VTAM LUs).
- Specify different UP / DOWN status names. For example, for objects of a certain type ONLINE / OFFLINE or ACTIVE / INACTIVE may be appropriate.
- Define different columns in the Object databases. For example, a column for started task objects containing the open reply ID of the started task (if any). A reply ID is meaningful only for a started task but is meaningless for a VTAM LU.

Objects defined in each Source Object database are copied to Working Object databases when CONTROL-O / COSMOS is initialized.

In a multi-CPU environment, a single Source Object database (stored on disk) can be used to generate a Working Object database (in memory only) in each CPU. This simplifies object management by enabling management of Object databases in multiple CPUs using a single central file (meaning, a Source Object database).

Creating and Maintaining the Prerequisite Database

The CONTROL-O / COSMOS Prerequisite database contains prerequisites for CONTROL-O / COSMOS-controlled objects. Only one CONTROL-O / COSMOS Prerequisite database can be defined.

When CONTROL-O / COSMOS determines that an object's current status does not match its desired status, CONTROL-O / COSMOS checks the prerequisites for that object in the desired status.

The CONTROL-O / COSMOS Prerequisite database determines whether or not an object can be brought to its desired status, and in this way controls:

- Sequence in which objects are activated during IPL.
- Sequence in which objects are deactivated during a shutdown.

Because CONTROL-O / COSMOS can manage more than one Object database, prerequisites can describe dependencies between objects in the same or different Object databases.

Prerequisites and Inverse Prerequisites

A standard CONTROL-O / COSMOS prerequisite specifies a particular object that must be up before another specific object can be brought up.

Unless indicated otherwise, CONTROL-O / COSMOS automatically applies the inverse relationship of all prerequisites as well. When bringing down objects, inverse application of CONTROL-O / COSMOS prerequisites helps ensure that objects are brought down in a logical order during system shutdown.

For example, if object A can only be brought up when object B is already up, object B can only be brought down if object A is already down.

Although application of the inverse relationship is the default, exceptions to this can be specified in the optional WAY field (described below).

Columns of the Prerequisite Database

The Prerequisite database must contain the following columns:

Table 16 Prerequisite Database Columns

Column	Description
FMOBJ	From Object. The object that can only be brought up if the object specified in TOOBJ is already up.
TOOBJ	To Object. The object that must be up before the object specified in FMOBJ can be brought up.
FMDB	From database. The Working Object database containing the FMOBJ object. A value is specified for this column only if the FMOBJ and TOOBJ objects are in different Working Object databases.
TODB	To database. The Working Object database containing the TOOBJ object. A value is specified for this column only if the FMOBJ and TOOBJ objects are in different Working Object databases.The following reserved prerequisite database field is optional:
WAY	Dependency relationships between objects at startup and shutdown. When a dependency relationship between two objects exists at startup, the inverse dependency normally exists at shutdown. Valid values: ■ blank—Both the normal and inverse prerequisite conditions exist. Default. This also applies if the column in omitted. ■ UP—Only the normal (startup) prerequisite exists. ■ DOWN—Only the shutdown (inverse) prerequisite exists.

The following screen segment contains sample content of a Prerequisite database (as displayed in the Variable Database Definition screen):

Figure 12 Prerequisite Database Example

```
     ROW       FMOBJ     TOOBJ     FMDB      TODB       WAY

     00007000  TCP       VTAM                           UP
     00008000  CICT      APLCICT                        UP
     00010000  IMS       JES2                           UP
     00011000  VTAM      JES2
     00012000  TCP       APLTCP    COSSTCOB  COSVTMOB
     00013000  CICP      VTAM                           DOWN
     00014000  CTD       JES2                           DOWN
```

CONTROL-O / COSMOS checks prerequisite dependency relationships between objects, for UP / DOWN, DOWN / UP or equivalent statuses (for example, ONLINE / OFFLINE, described in “Generic Statuses” below). Other statuses are not checked for prerequisites.

Creating and Maintaining Method Databases

Method rules are CONTROL-O rules that are invoked to change the status of an object. To determine which Method rule to invoke, CONTROL-O / COSMOS searches the Method database. Rows of the Method database contain information (methods) indicating which rule performs the change of an object from its current status to its desired status.

NOTE

Multiple Method databases can be specified. A Method database can be associated with more than one Object database. However, each Object database can be associated with only one Method database.

Each Method database must contain the following columns:

Table 17 Method Database Columns

Column	Description
CURRENT	Current status of the object.
DESIRED	Desired state of the object.
OBJDB	Name of the Object database in which the object is defined.
METCLASS	Name of an object or object class.
METHOD	Name of the Method rule to be triggered, and arguments (if any) to be passed to the Method rule. A Method rule must be triggered using an ON RULE statement. The name specified in the Method column must match the name in the ON RULE statement.

The following screen segment contains sample content of a Method database (as displayed in the Variable Database Definition screen):

Figure 13 Method Database Screen Example

```
     ROW       CURRENT   DESIRED   OBJDB     METCLASS  METHOD

     00001000  DOWN      UP                            COSMET01
     00002000  UP        DOWN                          COSMET04
     00003000  UNKNOWN                                 COSMET03
     00004000  DOWN      UP        COSVTMOB            COSMET11
     00005000  UP        DOWN      COSVTMOB            COSMET12
     00006000  UNKNOWN             COSVTMOB            COSMET13
```

CONTROL-O / COSMOS compares the information in each row of the Method database with the attributes of the object whose status is to be changed.

Attributes of a method may match the object’s attributes exactly, or a partial match may occur. Furthermore, multiple methods can match object attributes to varying degrees. Therefore, an algorithm is used rate (assign a score to) each method. Based on this rating, the method that best matches the attributes of the object to be changed is invoked. For more information, see “Generic Statuses” on page 57, and “Method Selection Logic” on page 59.

Generic Statuses

The most common statuses for CONTROL-O / COSMOS-controlled objects, are UP, DOWN and UNKNOWN. Different terminology may be appropriate for some object types. For this reason, CONTROL-O / COSMOS allows specification of different UP, DOWN and UNKNOWN values in member COSMOLST for each Object database. For example, objects in a certain database may have a status of ONLINE or ACTIVE instead of UP, a value of OFFLINE or INACTIVE instead of DOWN, and a value of NOTFOUND instead of UNKNOWN.

When comparing an object’s current and desired statuses with rows in the Method database, CONTROL-O / COSMOS first attempts to match the actual status name (for example, ONLINE or OFFLINE). However if UP, DOWN or UNKNOWN is specified in the Method database, this results in return of a partial match. CONTROL-O / COSMOS also matches any status beginning with the characters UP and DOWN with statuses UP and DOWN respectively.

- UP is a generic status which matches any of the following:
  - A status of UP.
  - A status beginning with the characters UP (for example, UPLOADED).
  - An UP value (for example, ONLINE or ACTIVE) specified for the Object database in member COSMOLST.
- DOWN is a generic status which matches any of the following:
  - A status of DOWN.
  - A status beginning with the characters DOWN (for example, DOWNLOAD).
  - A DOWN value (for example, OFFLINE or INACTIVE) specified for the Object database in member COSMOLST.
- UNKNOWN is a generic status which matches any of the following:
  - A status of UNKNOWN.
  - An UNKNOWN value (for example, NOTFOUND) specified for the Object database in member COSMOLST.

During the search for a method to change an object’s status, an exact match of status names (for example, ONLINE to ONLINE) is assigned a higher priority than a match to a generic status name (for example, ONLINE to UP). For more information, see “Method Selection Logic” on page 59.

Adding a New Method to CONTROL-O/COSMOS

A sample Method database and Method rules are supplied with CONTROL-O / COSMOS. However, additional methods are needed to implement CONTROL-O / COSMOS management of objects in your operating environment.

A method must exist for each action CONTROL-O / COSMOS may be required to perform. If a new object is defined, you must verify that methods for bringing that object UP and DOWN exist. If they do not, you must create them.

Creation of a new method involves:

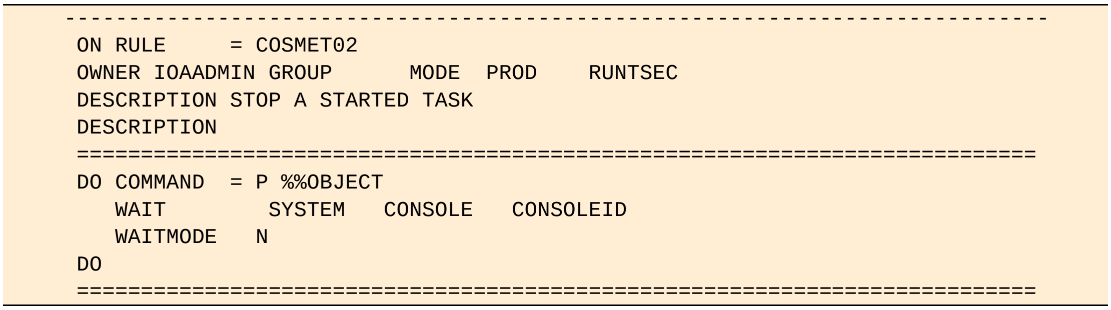
- Definition of CONTROL-O rules (meaning, Method rules) to perform the necessary actions.
- Addition of rows in the relevant Method databases that reference the Method rules and indicate when and for which objects they are used.

When a CONTROL-O / COSMOS method is selected, CONTROL-O / COSMOS rule COSFSMAC is triggered first to automatically point to the object to be modified. Therefore, Method rules need only indicate actions to perform on the actual object (resource) to be modified. For more information, see “COSFSMAC Rule” on page 65.

Example

The following sample Method rule can be used to stop started tasks:

Figure 14 Sample Method Rule



```
-----------------------------------------------------------------------------
 ON RULE     = COSMET02
 OWNER IOAADMIN GROUP      MODE  PROD    RUNTSEC
 DESCRIPTION STOP A STARTED TASK
 DESCRIPTION
 ===========================================================================
 DO COMMAND  = P %%OBJECT
    WAIT        SYSTEM   CONSOLE   CONSOLEID
    WAITMODE    N
 DO
 ===========================================================================
```

After creating a new Method rule, add an appropriate row in the relevant Method database. This is accomplished using the Variable Database Definition screen (screen IV).

NOTE

Each variable in the current row can be accessed using the name of the relevant column prefixed by %%. For example, %%MODE refers to the value in the MODE column of the current row in the Object database.

To enable immediate use of a new Method rule, issue command COSBOUNZ, described on page 79. This command brings CONTROL-O/COSMOS down, reloads all CONTROL-O/COSMOS databases, and restarts CONTROL-O/COSMOS.

For more information about the Variable Database Definition facility, see the online facilities chapter of the *CONTROL-O User Guide*.

Method Selection Logic

CONTROL-O/COSMOS compares attributes of the object whose status is to be changed with rows in the Method database. CONTROL-O/COSMOS attempts to match current status, desired status, object name, and object class with information specified for the various methods in the Method database.

- Each row in the Method database is assigned a score.
- Exact matches of object and method attributes are assigned the highest score.
- Partial matches (for example, matches with a generic status) are assigned an intermediate score.
- In certain columns, null can be specified. If an attribute of the object is compared with a method attribute specified as a null value, it is assigned a low score.
- Mismatches (for example, a comparison of status UP and status DOWN) result in rejection of the current method. CONTROL-O/COSMOS continues with the next row in the Method database.

Below is a list of the columns in the Method database and how they are compared with attributes of the object under consideration. All columns listed below are analyzed. CONTROL-O/COSMOS decides which method to invoke only after analyzing all rows in the Method database.

Each row in the Method database is assigned a score depending on how well it matches the description of the object to be modified. The Method rule specified in the row with the highest score is invoked. If no method matches the object under consideration, no action is performed.

The Method database columns are listed below in order of importance. For example, the first column (CURRENT) is assigned a more significant score than the second (DESIRED).

The following Method database columns are analyzed:

- CURRENT
  Compared with the current status of the object under consideration.
  - If the current status of the object matches the CURRENT column of the Method database row, the row is assigned a high score.
  - If the current status of the object matches a generic status specified in the CURRENT column in the Method database row, the row is assigned an intermediate score.
  - If the current status of the object does not match the CURRENT column, the row is rejected, and CONTROL-O / COSMOS continues with the next row in the Method database.
- DESIRED
  Compared with the desired status of the object under consideration.
  - If the desired status of the object matches the DESIRED column of the Method database row, the row is assigned a high score.
  - If the desired status of the object matches a generic status specified in the DESIRED column in the Method database row, the row is assigned an intermediate score.
  - If a null value is specified in the Method database row, the row is assigned a low score.
  - If the desired status of the object does not match the DESIRED column, the row is rejected, and CONTROL-O / COSMOS continues with the next row in the Method database.
- OBJDB
  Compared with the name of the Working Object database containing the object under consideration.
  - If the value in this column matches the name of the Working Object database, the row is assigned a high score.
  - If a null value is specified in this column, a low score is assigned.
  - If a value is specified in this column and it does not match the name of the object database, the row is rejected, and CONTROL-O / COSMOS continues with the next row in the Method database.

- METCLASS
  Compared with the name and class of the object under consideration.
  - If the name of the object matches the value in this column, the row is assigned a high score.
  - If the name of the object does not match the METCLASS column but the name of the object class does match, the row is assigned an intermediate score.
  - If a null value is specified in this column, the row is assigned a low score.
  - If a value is specified in this column and it does not match the name of the object or the object class, the row is rejected, and CONTROL-O / COSMOS continues with the next row in the Method database.

The scores assigned to matches in each column of a row are added together to determine its total score.

The rule named in the METHOD column of the row with the highest total score is invoked to process the object.

If more than one row is assigned the highest total score (that is, if there is a tie), the first- appearing row with the highest total score is chosen.

As already mentioned, if no row match was found (meaning, all rows were rejected), no action is performed on the object being considered.

NOTE

Rows rejected at any stage in the comparison are not considered candidates for processing the object.

The following table shows the possible combinations of matches of information that can invoke a Method rule. These combinations are listed in order of their scores (that is, row number 1 is the highest score, 2 the second highest, and so on).

Table 18 Method Rule Invocations (part 1 of 2)

	Current		Desired			OBJDB		METCLASS		
	Exact Match	Generic Match	Exact Match	Generic Match	Null	Exact Match	Null	Object Name	Object Class	Null
1	Yes		¸			¸		¸		
2	Yes		¸			¸			¸	
3	Yes		¸			¸				Yes
4	Yes		¸				¸	Yes		
5	Yes		¸				¸		¸	

Table 18 Method Rule Invocations (part 2 of 2)

	Current		Desired			OBJDB		METCLASS		
	Exact Match	Generic Match	Exact Match	Generic Match	Null	Exact Match	Null	Object Name	Object Class	Null
6	Yes		¸				¸			Yes
7	Yes			¸		¸		¸		
8	Yes			¸		¸			¸	
9	Yes			¸		¸				Yes
10	Yes			¸			¸	Yes		
11	Yes			¸			¸		¸	
12	Yes			¸			¸			Yes
13	Yes				¸	Yes		¸		
14	Yes				¸	Yes			¸	
15	Yes				¸	Yes				Yes
16	Yes				¸		¸	Yes		
17	Yes				¸		¸		¸	
18	Yes				¸		¸			Yes
19		¸	Yes			¸		¸		
20		¸	Yes			¸			¸	
21		¸	Yes			¸				Yes
22		¸	Yes				¸	Yes		
23		¸	Yes				¸		¸	
24		¸	Yes				¸			Yes
25		¸		¸		¸		¸		
26		¸		¸		¸			¸	
27		¸		¸		¸				Yes
28		¸		¸			¸	Yes		
29		¸		¸			¸		¸	
30		¸		¸			¸			Yes
31		¸			¸	Yes		¸		
32		¸			¸	Yes			¸	
33		¸			¸	Yes				Yes
34		¸			¸		¸	Yes		
35		¸			¸		¸		¸	
36		¸			¸		¸			Yes

CONTROL-O/COSMOS Rules

Many CONTROL-O / COSMOS tasks are performed using specially defined CONTROL-O rules. Throughout this chapter, method rules have been mentioned as the rules that perform the actions that bring an object to its desired status. However, keep in mind the following points:

- Method rules do not work alone. They work together with another rule, called COSFSMAC.
- There are other sets of rules (besides COSFSMAC and Method rules) that perform functions other than bringing an object to its desired status.

The following rules and types of rules work together according to function. They are described in detail in this section.

- COSFSMAC Rule and Method Rules

  These rules perform the changes required to bring an object to its desired status.

- CONTROL-O / COSMOS Initialization Rules

  These rules are used in CONTROL-O / COSMOS initialization.

- System Event Detection Rules, The COSELECT rule, and Filter and Action Rules (including Generic Action rule COSACT99)

  These rules work together to detect system events (for example, the starting up or bringing down of a started task) and update the appropriate Working Object database with the information.

- User-Defined Action Rule COSACT98

  This rule is triggered by option R in the Object Status screen, and can be used to define a customized action to be performed when this option is chosen.

- Miscellaneous Rules

  Several rules that perform miscellaneous functions.

A list of local and Global variables used by CONTROL-O / COSMOS rules is provided following the rule descriptions.

General Organization of the CONTROL-O/COSMOS Rules

CONTROL-O/COSMOS is implemented by a CONTROL-O subtask whose main objective is to check the status of the controlled objects, and to issue a message when an action is required. These messages cause CONTROL-O/COSMOS rules to be triggered, which work to balance the system as necessary.

As of version 6.1.00, the CONTROL-O/COSMOS table has been split into two separate tables, $COSMOSO and $COSMOSU. These tables are stored in the IOA*highlevel*.IOAENV and CTO*highlevel*.RULES libraries, respectively.

- $COSMOSO table contains those rules provided and maintained by BMC Software. This table is maintainable through SMP/E fixes.

- $COSMOSU table contains those rules that you can modify, and is therefore not maintained by SMP/E fixes.

COSRUL Rule

As of version 6.1.00, rules are no longer directly executed by DO RULE statements, but instead through the COSRUL rule. For example, the syntax of statements such as

```
DO RULE=COSELECT ....
```

has been changed to

```
DO RULE=COSRUL COSELECT ....
```

In addition to calling a specified CONTROL-O/COSMOS rule, COSRUL also determines whether to preprocess, postprocess, or override that rule by using one of the following special AutoEdit variables:

- %%PRE_*rulename*
- %%PST_*rulename*
- %%OVR_*rulename*

where *rulename* is the name of the relevant CONTROL-O/COSMOS rule.

In this way, you can add additional statements both before and after those provided by BMC Software and, if necessary, completely override the original rule.

The following statements, which are found in the USRCLEAR rule of the $COSMOSU table, cause the user-defined rules PREMET01, PSTMET03, and OVRMET04 in the $COSMOSU table to preprocess, postprocess, and override the COSMET01, COSMET03, and COSMET04 rules in the $COSMOSO table, respectively:

```
/*
/* PREMET01 TO EXECUTE BEFORE COSMET01
/*
DO SET=%%PRE_COSMET01=PREMET01
/*
/* PSTMET03 TO EXECUTE AFTER  COSMET03
/*
DO SET=%%POST_COSMET03=PSTMET03
/*
/* OVRMET04 TO OVERRIDE COSMET04
DO SET=%%OVER_COSMET04=OVRMET04
```

For example, when you execute the command

```
DO RULE=COSRUL COSMET01
```

COSRUL uses the information in the first DO SET statement to execute the PREMET01 rule before calling the COSMET01 rule.

COSFSMAC Rule

When CONTROL-O / COSMOS determines that an object is not in its desired status, it searches for a method to set the object to its desired status. When an appropriate method is found, the following internal message is issued:

```
COSFSMAC dbname row rulename parms
```

where

- *dbn*ame – Name of the Working Object database containing the object to be modified.
- *row* – Number of the database row describing the object to be modified.
- *rulename* – Name of the Method rule to be used to modify the object.
- *parms* – Parameters (if any) to be passed to the Method rule.

This message triggers rule COSFSMAC that:

- Specifies the Object database containing the object to be modified as the current Global Variable pool (Reserved User-defined variable %%$GLOBAL).

- Specifies the row describing the object to be modified as the current database row (Reserved User-defined variable %%$DBROW).

- Triggers the specified Method rule.

Since the COSFSMAC rule performs the above routine tasks, it simplifies coding of Method rules. Method rules assume that when they are triggered, the Global variable pool and database row are already set. Only actions to be performed (for example, issuing a START command) need to be included in the Method rule.

Method Rules

Method rules define actions that change the status of a CONTROL-O / COSMOS-controlled object.

Method rules must begin with an ON RULE statement. They are triggered by a DO RULE statement in the COSFSMAC rule. The rule name specified in the ON RULE statement of the Method rule must be the name specified in the METHOD column of the Method database.

When CONTROL-O / COSMOS determines that an object's status does not match its desired status, it searches the Method database for the name of a method to change the status of the object. When a matching row is found in the Method database, the rule specified in the METHOD column of the row is triggered (using the COSFSMAC rule).

Sample Method rules (with a prefix of COSMET) are provided with CONTROL-O / COSMOS. These Method rules can be customized and additional Method rules can be defined to conform with site requirements.

CONTROL-O/COSMOS Initialization Rules

This topic describes the COSMOLST and Object database initialization rules.

COSMOLST Variable Initialization Rules

When CONTROL-O / COSMOS is initialized, the contents of each record in member COSMOLST are displayed in a message. These messages trigger the following rules to process COSMOLST information and to set Global variables used by CONTROL-O / COSMOS rules:

Table 19 Initialization Rules

Rule	Description
COSFSMRH	Processes the Header record of member COSMOLST.
COSFSMRT	Processes Object database records of member COSMOLST.
COSFSMRC	Processes CPU records of member COSMOLST.

For more information about the Global variables used by CONTROL-O / COSMOS, see "AutoEdit Variables Used by CONTROL-O / COSMOS Rules" on page 74.

Object Database Initialization

The following rules handle initialization of the CONTROL-O / COSMOS Object databases.

- COSFSMIN
  Handles initialization of Object databases. During initialization of CONTROL-O / COSMOS, this rule copies records from the Source Object database to the Working Object database and specifies appropriate values for the various columns in the Working Object database.

  The rule also sets the current status of all objects to UNKNOWN. Therefore, the next time CONTROL-O / COSMOS scans the Object database, a rule is triggered to determine the current status of each object.

- COSFSMIC
  Resets certain Global variables after all CONTROL-O / COSMOS Object databases have been initialized. This rule can be modified to perform additional actions that are required after CONTROL-O / COSMOS initialization is completed.

CONTROL-O/COSMOS Object Update Rules

System event detection rules detect system events that affect objects controlled by CONTROL-O / COSMOS (for example, the starting up or bringing down of a started task).

After a system event detection rule detects a system event, the rule passes parameters to, and triggers, the CONTROL-O / COSMOS COSELECT rule. The parameters passed to the COSELECT rule are the names of a Filter rule and an Action rule. The COSELECT rule uses the Filter and Action rules to register the event in the appropriate Working Object database object. For more information, see "COSELECT Rule".

General Rules

Every event-detection rule triggers a General rule that handles events of the appropriate type (for example, starting a started task). The General rules determine which arguments to pass to the COSELECT rule.

The following General rules are supplied with CONTROL-O / COSMOS:

Table 20 CONTROL-O/COSMOS General Rules

Rule	Description
COSSTCUP	Handles events that indicate that a started task was activated (for example, message IST020I VTAM INITIALIZATION COMPLETE). Relevant messages are intercepted by rules that trigger rule COSSTCUP to pass appropriate arguments to the COSELECT rule.
COSSTCDW	Handles events that indicate that a started task was brought down. Relevant messages are intercepted by rules that trigger rule COSSTCDW to pass appropriate arguments to the COSELECT rule.
The following General rules are triggered directly by events in the system:	
START	Handles START commands issued for objects controlled by CONTROL-O / COSMOS. This rule triggers the COSELECT rule to update the desired status of the object in the Working Object database. CONTROL-O / COSMOS detects this change the next time it scans the Object database.
STOP	Handles STOP commands issued for objects controlled by CONTROL-O / COSMOS. This rule triggers the COSELECT rule to update the desired status of the object in the Working Object database. CONTROL-O / COSMOS detects this change the next time it scans the Object database.
IEF403I	Triggered by message IEF403I, indicating that a started task is starting. This rule triggers the COSELECT rule, which searches for the object and triggers a rule that changes the current status of the object to STARTING.
*	Triggered by all job end events (ON JOBEND =*). This rule triggers the COSELECT rule that searches for the relevant object and, if found, changes the current status of the object to DOWN.
*	Triggered by all messages in the system (ON MESSAGE set to *). If the detected message is a WTOR message, this rule triggers the COSELECT rule to update the REPLYID column for the relevant object.
$HASP310	Triggered by message $HASP310, indicating that a started task was ended due to a memory problem. The COSELECT rule is triggered to search for the started task name that appeared in the message and, if, found, changes the current status of the relevant object to DOWN.

COSELECT Rule

The COSELECT rule is triggered by other CONTROL-O / COSMOS rules to search for Object database rows on which to perform a specified action. The names of a Filter rule and an Action rule are supplied as parameters to the COSELECT rule.

The COSELECT rule performs the following actions:

1. The COSELECT rule applies the specified Filter rule to each row of the specified Object databases.

   If the current row is selected by the Filter rule, the specified Action rule is triggered to perform an action on the current row.

2. The COSELECT rule increments the row number (%%$DBROW) and applies the Filter rule to the next row of the Object database.

Filter rules and Action rules are described later in this chapter.

The DO RULE statement that triggers the COSELECT rule supplies the following parameters:

```
COSELECT database filter action
```

where

- *database* is the name of the Working Object database to scan. Valid values:
  - — *dbname* – The name of the database to scan.
  - — ALL – Scan all databases.
- *filter* is the name of the Filter rule applied to each database row scanned by the COSELECT rule.
- *action* is the name of the Action rule to apply to database rows chosen by the Filter rule.

Example

Assume that the following statement is used to trigger the COSELECT rule:

```
DO RULE=COSELECT DATABASE1 FILTER01 ACTION01
```

Rule FILTER01:

```
IF       %%CURRENT NE %%DESIRED
DO SET      = %%COS_SELECTED = YES
ENDIF
```

Rule ACTION01:

```
DO SHOUT    = TO OPER              URGENCY R SYSTEM
   MESSAGE CONTROL-O/COSMOS -
%%CPU/%%OBJECT/%%CURRENT/%%DESIRED/%%MODE/%%TEXT
```

A Shout message is issued for each row of DATABASE1 in which the CURRENT column and the DESIRED column do not match.

Filter Rules

Filter rules are triggered by the COSELECT rule, described in “COSELECT Rule” on page 69, to determine whether or not to perform an action on the current row in an Object database.

If an action must be performed on the current row, the Filter rule sets variable %%COS_SELECTED to YES. The COSELECT rule detects this and triggers the appropriate Action rule.

Sample Filter rules (with a prefix of COSFLT) are provided with CONTROL-O/COSMOS. These filters can be customized and additional Filter rules can be defined to conform with your site’s requirements.

Action Rules

Action rules are triggered by the COSELECT rule when a Filter rule determines that an action is to be performed on the current row in an Object database. The name of the Action rule to be applied is passed as a parameter to the COSELECT rule by the General rule that handled the system event.

Sample Action rules (with a prefix of COSACT) are provided with CONTROL-O/COSMOS. These actions can be customized and additional Action rules can be defined to conform with your site’s requirements.

COSACT99 Rule

The COSACT99 rule is a generic Action rule that can be used to define tasks to be performed subsequent to execution of each Action rule. Most Action rules trigger the COSACT99 rule before returning control to the COSELECT rule.

Sample rule COSACT99 that is supplied with CONTROL-O/COSMOS checks if debug information is displayed. Additional actions can be specified in this rule to conform with your sites requirements.

User-Defined Action Rule COSACT98

The COSACT98 rule is triggered when option R (User) is specified on the CONTROL-O / COSMOS Object Status screen.

Sample the COSACT98 rule supplied with CONTROL-O / COSMOS issues a Shout message describing the object for which the option was specified.

This rule can be customized to perform a user-defined action (for example, one not available using the predefined options in the Object Status screen).

Miscellaneous Rules

Consider also the following rules when implementing CONTROL-O / COSMOS at your site:

Table 21 Miscellaneous CONTROL-O/COSMOS Rules (part 1 of 3)

Rule	Description
COSBROAD	Triggered during CONTROL-O / COSMOS initialization and termination. When triggered, this rule broadcasts the current status of CONTROL-O / COSMOS from the computer where CONTROL-O / COSMOS was initialized or terminated to all other computers that are running CONTROL-O / COSMOS in a Sysplex. This rule is used in combination with the COSBRREC rule, described in this table.
COSBRREC	Triggered when a COSBROAD rule broadcast, described in this table, is received. When triggered, the COSBRREC rule performs the following actions: 1. Receives the change in CONTROL-O / COSMOS status broadcast by another computer in the Sysplex. 2. Updates the respective %%COSTASK_%%*computer* variable, where *computer* is the name of the broadcasting computer. If the variable for the broadcasting computer does not exist, the rule creates it Transmits the status of the reporting computer to the computer that broadcast the status change, which in turn updates or creates the respective variable for the reporting computer

Table 21 Miscellaneous CONTROL-O/COSMOS Rules (part 2 of 3)

Rule	Description
COSCLEAR	Triggered when the $COSMOSO table is loaded. This ON EVENT rule initializes (clears) some global variables used by CONTROL-O/COSMOS. These variables are kept in a temporary AutoEdit pool named COSWORKV during initialization.
COSFSMPR	Triggered when the COSFSMPR message is issued. Display of this message is optional, and depends on the flag in the COSMOLST member, described on page 90. The COSFSMPR message indicates that CONTROL-O/COSMOS found a prerequisite that has not been satisfied. The format of the message is `COSFSMPR obj_db obj_row obj_name prq_db prq_row prq_name abcd` where ■ *obj_db, obj_row, obj_name* are the database, row, and name of the object analyzed by CONTROL-O/COSMOS ■ *prq_db, prq_row, prq_name* are the database, row, and name of the missing prerequisite ■ *abcd* is a 4-character string, comprised of the following: *a*—the current status of the object *b*—the desired status of the object *c*—the current status of the prerequisite object *d*—the desired status of the prerequisite object
COSFSMTX	Triggered each time the value of %%STATUS changes. Updates values in the STATUS column of Working Object databases (viewed using the Object Status screen) with the result of the last scan CONTROL-O/COSMOS performed on the Object database. For more information about the Object Status screen, see page 29.

Table 21 Miscellaneous CONTROL-O/COSMOS Rules (part 3 of 3)

Rule	Description
COSINTRL	Triggered by the COSCLEAR rule, described on page 72, at CONTROL-O/COSMOS initialization. The COSINTRL rule sets one AutoEdit override variable (%%OVER_*rulename*) for each rule name contained in the %COSMOSO table. In this way, CONTROL-O/COSMOS can determine whether a rule is stored in either the $COSMOSO or $COSMOSU table. For example, the statement `DO SET=%%OVER_COSMET12=INTERNAL GLOBAL Y` specifies that, unless explicitly overridden by the user, the COSMET12 rule is INTERNAL. This means that the COSMET12 rule in the $COSMOSO member is used.
	Note: The statement `DO SET=%%OVER_COSMET04=OVRMET04 GLOBAL Y` in the USRCLEAR rule is an example of how a user can override internal rules.
COSSTCHK	Triggered at specified intervals to check the status of started tasks controlled by CONTROL-O/COSMOS. If a started task's current status does not match the status in the Object database, a Shout message (specified in rule COSACT85) is issued to notify the user of the inconsistency.
CTO603I	Triggered by options specified in the Object Status screen. This rule triggers an Action rule that performs the action requested by the user.
CTO613I	Triggered by options specified in the Database Status screen. This rule triggers an Action rule that performs the action requested by the user.
CTO658I and CTO659I	Triggered by messages issued by the CTOCTI utility. These rules are used by the SYSIMAGE facility to build an Object database that creates an image of the current environment. For more information, see "SYSIMAGE Facility – Automatic Generation of an Object Database" on page 96, and the discussion of the CTOCTI utility in the *INCONTROL for z/OS Utilities Guide.*
USRCLEAR	Overrides some global rules used by CONTROL-O/COSMOS, when modified by the user, which were initially set by the COSCLEAR rule.
VARCLEAR	Triggered when the $COSMOSU member is loaded. This rule initializes variables, directly and by calling other rules, that are required for COSMOS implementation and rule dispatching. It also saves the library name and member from where the $COSMOSU table was loaded.

AutoEdit Variables Used by CONTROL-O/COSMOS Rules

The variables described in the following tables are used by CONTROL-O / COSMOS rules.

- Local variables are used by the COSELECT rule, and related rules, to keep track of information during the run of a COSELECT rule.
- Most Global variables listed below are set by CONTROL-O / COSMOS initialization rules. These variables are stored in Global variable pool COSWORKV.

For more information about how these variables are used, see rule table CONTROL-O / COSMOS in the CONTROL-O Rule library.

Local Variables

Table 22 Local Variables

Variable	Description
%%COS_IROW	Current row being processed by the COSELECT rule.
%%COS_ODB	Current Working Object database being processed by COSELECT.
%%COS_SELECTED	Whether (or not) the database row currently being processed by the COSELECT rule is to be processed by the specified Action rule. Valid values are: ■ YES—Trigger the Action rule for the current database row. ■ NO—Do not trigger the Action rule for the current database row.
%%COS_SELROW	Number of rows selected by the current Filter rule used by the COSELECT rule.
%%COSDOWN	DOWN value specified for the database currently processed by the COSELECT rule (for example, DOWN, INACTIVE or OFFLINE).
%%COSUNKN	UNKNOWN value specified for the database currently processed by the COSELECT rule (for example, UNKNOWN or NOTFOUND).
%%COSUP	UP value specified for the database currently processed by the COSELECT rule (for example, UP, ACTIVE or ONLINE) Global Variables.
%%SRCDB	Current Source Object database being processed by COSELECT.

Global Variables

Table 23 Global Variables (part 1 of 2)

Variable	Description
%%COS_OPT	The startup option most recently selected by the operator during CONTROL-O / COSMOS initialization.
%%COSCPUNUM	The number of CPUs specified in member COSMOLST.
%%COSDB_*n*	The name of the nth Working Object database.
%%COSDBNUM	The number of Object databases being used by CONTROL-O / COSMOS.
%%COSDEMOMODE	Whether CONTROL-O / COSMOS demonstration databases are in use. Valid values are YES and NO.
%%COSDOWN_*n*	The DOWN value for the *n*th Object database.
%%COSFFLGS	The CONTROL-O / COSMOS subtask flags (for future use).
%%COSINX_*dbname*	The serial number (for example, 1, 2, or 3) of database *dbname*.
%%COSLIBRARY	Unchangeable rules library. Default: IOA*prefix*.IOAENV.
%%COSMEM	Unchangeable rules member. Default: $COSMOSO.
%%COSMT_*n*	The name of the *n*th Method database.
%%COSOFLGS	The operation flags (for future use).
%%COSPREFIX	The prefix of CONTROL-O STCs (%PROCPRFO). Used to accept, reject, or prefix the commands submitted to or issued from this IOA installation.
%%COSPRETB	The name of the CONTROL-O / COSMOS Prerequisite database.
%%COSSRCDB_*n*	The name of the nth Source Object database.
%%COSSTC	The name of the Object database containing definitions of started tasks controlled by CONTROL-O / COSMOS.
%%COSTASKW	The desired status of the CONTROL-O / COSMOS facility. Rule USRCLEAR, which is triggered at CONTROL-O startup, sets this variable to DOWN. If you want CONTROL-O / COSMOS to be initialized during CONTROL-O startup, edit this rule and set the variable to UP.
%%COSUNKN_*n*	The UNKNOWN value for the *n*th Object database.
%%COSUP_*n*	The UP value for the *n*th Object database.
%%COSYSPLEX	Whether CONTROL-O / COSMOS works in a Sysplex environment, and what type of database it uses. Valid values are: ■ YES – CONTROL-O / COSMOS is working in a Sysplex environment and is using a Syslpex (XAE type 1) database. ■ NO – Either CONTROL-O / COSMOS is not working in a Sysplex environment, or it is, but is not using a Sysplex database.
%%CPU_*n*	The SMF ID or SYSTEM ID of the *n*th CPU.

Table 23 Global Variables (part 2 of 2)

Variable	Description
%%CTO659I	The number of CTO659I messages processed by the SYSIMAGE facility.
%%USRLIBRARY	Changeable rules library. Default: CTO*prefix*.RULES.
%%USRMEM	Changeable rules member. Default: $COSMOSU.

CONTROL-O/COSMOS Commands

Several commands are provided with the CONTROL-O / COSMOS facility. These commands are divided into the following groups:

- commands that affect the operation of CONTROL-O / COSMOS
- commands that perform actions on CONTROL-O / COSMOS-controlled objects, many of which perform actions that can also be performed using CONTROL-O / COSMOS online screens

Except where noted, CONTROL-O / COSMOS commands have no special parameters or syntax. To use them, type the command where appropriate on the current screen and press **Enter**.

This section describes each CONTROL-O / COSMOS command and the rules that they trigger.

NOTE

The following are new characteristics of CONTROL-O / COSMOS commands as of version 6.1.00:

- Prior to version 6.1.00, all CONTROL-O / COSMOS commands began with the prefix “COS” (for example, COSINI and COSUP). As of version 6.1.00, CONTROL-O / COSMOS those commands that are implemented by CONTROL-O / COSMOS rules use prefixes that are defined during CONTROL-O installation. For further details about these prefixes, see “Supporting Multiple CONTROL-O / COSMOS Instances” on page 3-85.
- As of version 6.1.00, CONTROL-O / COSMOS commands are not implemented by ON COMMAND rules, but instead are implemented by ON RULE rules.

*COS*INI Command – Starting CONTROL-O/COSMOS

NOTE

As of version 6.1.00, this command uses a predefined prefix in place of the original “COS” prefix. For more information about these prefixes, see “Supporting Multiple CONTROL-O / COSMOS Instances” on page 3-85.

Use this command to start CONTROL-O / COSMOS.

The following Shout messages are issued in response to the *COS*INI command:

Figure 15 *COS*INI Shout Messages

```
COSMOS -     THE FOLLOWING STARTING OPTIONS ARE AVAILABLE:
COSMOS -     1 - DESIRED = UP
COSMOS -     2 - DESIRED = DOWN
COSMOS -     3 - DESIRED = CURRENT (AND SET MODE FORCE_OK)
COSMOS -     4 - DESIRED = SOURCE DATABASE DESIRED COLUMN
COSMOS -     5 - DESIRED = SOURCE DATABASE ATIPL COLUMN
COSMOS -     6 - DESIRED = SOURCE DATABASE NOTIPL COLUMN
nn CONTROL-O/COSMOS - SELECT ONE OPTION
```

These messages indicate different options that can be used to determine startup status for CONTROL-O / COSMOS-controlled objects.

The option that is recommended depends on whether CONTROL-O / COSMOS will be responsible for starting other resources (objects) in the system.

- If CONTROL-O / COSMOS is started at IPL, option 5 is recommended. This option causes the desired status to be set according to the ATIPL column in the Source Object database.

- If CONTROL-O / COSMOS is started without an IPL, option 3 is recommended. Option 3 sets each object’s mode to FORCE_OK.

  When a CONTROL-O / COSMOS Object database is initialized, the current status of each object is set to UNKNOWN. The next time CONTROL-O / COSMOS scans the Object database, a rule is triggered to determine the current status.

- If startup option 3 (mode FORCE_OK) was specified at CONTROL-O / COSMOS startup, the desired status is modified to match the new current status. After current and desired statuses are set, objects can be assigned a mode of FREE to transfer control of the object status to CONTROL-O / COSMOS.

- If a startup option other than 3 was specified, the desired status for each object is set according to the specified option. The next time CONTROL-O / COSMOS scans the Object database, CONTROL-O / COSMOS takes steps to ensure that objects are brought to their desired statuses.

NOTE

The default startup option supplied with CONTROL-O / COSMOS is option 3 (that is, if no option is specified for 30 seconds this option is automatically implemented). This default can be modified (in the COSINIT0 rule) to conform with your site's requirements.

Command Workflow

The following steps are performed internally when you run the *COS*INI command:

1. The COSINIT0 rule is triggered. This rule issues a series of Shout messages describing available startup options, and one WTOR message telling the user to choose one of these options.

2. When a startup option is specified in response to the WTOR message, the option number is stored in an AutoEdit variable and the following command is issued:

   ```
   F CONTROLO,COSMOSSTART
   ```

3. During initialization, CONTROL-O / COSMOS issues internal messages in the following format:

   `COSFSMIN` *dbname*

   Where *dbname* is the name of a Working Object database to initialize.

4. The COSFSMIN rule is triggered by the message in step 3. This rule triggers the COSELECT rule.

5. The COSELECT rule applies Action rule COSACT70 to perform the following actions on each Object database row:

   — Copies necessary information from the Source Object database to the Working Object database.

   — Sets the CURRENT status of the object to UNKNOWN.

   — Sets the DESIRED status according to the option specified in response to the WTOR message described in Step 1.

*COS*TERM Command – Stopping CONTROL-O/COSMOS

NOTE

As of version 6.1.00, this command uses a predefined prefix in place of the original “COS” prefix. For more information about these prefixes, see “Supporting Multiple CONTROL-O / COSMOS Instances” on page 3-85.

Use this command to shut down the CONTROL-O / COSMOS facility.

This command shuts down the CONTROL-O / COSMOS facility without affecting the status of CONTROL-O / COSMOS-controlled objects.

*COS*TERM triggers the COSTERM rule, which issues the following operator command:

```
F CONTROLO,COSMOSSTOP
```

You can modify the COSTERM rule for added functionality.

*COS*BOUNZ Command – Bring Down and Restart CONTROL-O/COSMOS

NOTE

As of version 6.1.00, this command uses a predefined prefix in place of the original “COS” prefix. For more information about these prefixes, see “Supporting Multiple CONTROL-O / COSMOS Instances” on page 3-85.

This command bounces (recycles) the CONTROL-O / COSMOS facility by terminating CONTROL-O / COSMOS and then restarting it.

This command triggers the COSBOUNZ rule, which also reloads CONTROL-O / COSMOS databases, thereby refreshing all CONTROL-O / COSMOS information.

COSDB Command

NOTE

This command does not use a predefined prefix in place of the original “COS” prefix.

This command is used to change Object database or CONTROL-O / COSMOS mode, or to display information about Object databases. The command syntax is:

```
F CONTROLO,COSDB=dbname,cmd
```

where

- *dbname* is the name of the database on which the command is performed. mandatory. Valid values are:

name	The name of a Working Object database on which to perform the requested action
ALL	If the COSDB command is being used to change a mode, this value indicates that the mode of the CONTROL-O / COSMOS facility must be modified (and not a database specific mode).

If the COSDB command is being used to display information, this value indicates that information describing all Object databases is to be displayed.

- *cmd* is the action to be performed. Mandatory. Valid values are:

FREE	Assign a mode of FREE to CONTROL-O / COSMOS or to the specified Object database. ■ If you specified ALL for the command, the mode specified for each Object database determines how CONTROL-O / COSMOS-controlled objects are handled. ■ If you specified a specific database for the command, the mode specified for each object in the Object database determines how that object is managed by CONTROL-O / COSMOS.
HELD	Assign a mode of HELD to CONTROL-O / COSMOS or to the specified Object database. No action can be performed on objects, until the mode for CONTROL-O / COSMOS or the specified Object database is changed.
FORCE_OK	Assign a mode of FORCE_OK to CONTROL-O / COSMOS or to the specified Object database. The desired status of objects in the specified databases is modified so that it matches the current status.

NOPRE	Assign a mode of NOPRE to CONTROL-O / COSMOS or to the specified Object database. CONTROL-O / COSMOS does not check prerequisites before attempting to change the status of objects in the specified databases.
DISPLAY	Display information about all CONTROL-O / COSMOS databases or the specified database. Database name, mode and the name of the associated Method database are displayed for each Object database.

The COSDB command affects specified Object databases directly. No CONTROL-O / COSMOS rules are triggered to perform the actions requested using this command.

*COS*CMD Command

NOTE

As of version 6.1.00, this command uses a predefined prefix in place of the original “COS” prefix. For more information about these prefixes, see “Supporting Multiple CONTROL-O / COSMOS Instances” on page 3-85.

This command triggers the COSCMD rule, which changes the information in an Object database according to specified parameters.

The syntax for this command is:

```
COSCMD cmd dbname select-criteria
```

where

- *COS* is the special command prefix defined during CONTROL-O installation.
- *cmd*
  Action to be performed. Valid values are:

BOUNCE	Set desired status of specified objects to DOWN. When the current status becomes DOWN, set desired status to UP.
CURRDOWN	Set current status of specified objects to DOWN.
CURRUNKN	Set current status of specified objects to UNKNOWN.
CURRUP	Set current status of specified objects to UP.
DEBUGN	Set debug mode to N (No) for the specified objects.
DEBUGY	Set debug mode to Y (Yes) for the specified objects.
DESIDOWN	Set desired status of specified objects to DOWN.
DESIUP	Set desired status of specified objects to UP.

DISPALL	Display detailed information about specified objects. Multiple lines are used to describe each object.
DISPLAY	Display information about specified objects. One line of information is displayed for each object. Default.
MODEF	Set mode of specified objects to FREE.
MODEH	Set mode of specified objects to HELD.
MODEN	Set mode of specified objects to NOPRE.
MODEO	Set mode of specified objects to FORCE_OK.

- *dbname*
  Name of the database containing objects on which the command is to be performed. Optional. Valid values are:

name	Name a specific Object database.
ALL	Perform the requested action on objects described in all Object databases. Default.

- *select-criteria*
  Selection criteria that determine the objects to be processed by the command. Optional. Valid values are:

fieldname val	Where fieldname is the name of a column in the Object database, and val is a value that appears in that column. Mask characters cannot be specified in this field.
EXCEPTION	Indicates that the command is to process all objects whose current and desired statuses do not match.

A maximum of two selection criteria can be specified. If a value of EXCEPTION is specified, no other selection criteria can be specified.

If more than one selection criterion is specified, the selection criteria have an AND relationship.

If no selection criteria are specified, the requested action is performed on all objects defined in the specified database.

Examples

```
COSCMD
```

Display information about all objects that are controlled by CONTROL-O/COSMOS, describing one object per line.

```
COSCMD DISPLAY ALL EXCEPTION
```

Display one line of information about each object that is controlled by CONTROL-O/COSMOS, whose current and desired statuses do not match.

```
COSCMD DISPALL COSSTCOB OBJECT JES2
```

Display all information (that is, multiple lines) about objects in the COSSTCOB Object database that have a value of JES2 in the OBJECT column.

```
COSCMD DISPALL COSSTCOB GROUP  MYGROUP APPL MYAPPL
```

Display all information (that is, multiple lines) about objects in the COSSTCOB Object database that have a value of MYGROUP in the GROUP column, and a value of MYAPPL in the APPL column.

Command Workflow

The following actions are performed for each invocation of the *COS*CMD command:

1. The COSCMD rule is triggered. This rule determines the Filter and Action rules for the requested action and passes them to the COSELECT rule.

2. The COSELECT rule applies the specified Filter rule and Action rule to the specified Object databases.

For more information, see “COSELECT Rule” on page 69.

Shortcuts for the *COS*CMD Command

A number of shortcuts for the *COS*CMD command are supplied with CONTROL-O/COSMOS. Each of these shortcuts triggers a rule that invokes the *COS*CMD command with specific parameters.

NOTE

Shortcuts for the *COS*CMD command are initially defined to act only upon the demo databases supplied with CONTROL-O/COSMOS. To ensure that the *COS*CMD shortcuts act upon the Working Object databases in use at your site, some minor customization of the rules that invoke the shortcuts may be required.

The default settings for the *COS*UP and *COS*DOWN commands do not affect the demo Object database created using the SYSIMAGE facility (COSIMGSD). This setting helps prevent accidentally setting actual resources in your production environment to UP while you are working with demo databases.

*COS*UP — Start Up All CONTROL-O/COSMOS-Controlled Objects

NOTE

As of version 6.1.00, this command uses a predefined prefix in place of the original “COS” prefix. For more information about these prefixes, see “Supporting Multiple CONTROL-O / COSMOS Instances” on page 3-85.

This command sets the desired status of all objects that are controlled by CONTROL-O / COSMOS to UP, and their mode to FREE. The next time CONTROL-O / COSMOS scans the Object databases, it takes steps to ensure that objects not currently up are brought up (if prerequisites are satisfied). This command might typically be used to simulate an IPL or to test CONTROL-O / COSMOS operations.

The *COS*UP command invokes the COSUP rule, which issues the following command:

```
COSCMD DESIUP ALL
```

*COS*DOWN — Shut Down All CONTROL-O/COSMOS-Controlled Objects

NOTE

As of version 6.1.00, this command uses a predefined prefix in place of the original “COS” prefix. For more information about these prefixes, see “Supporting Multiple CONTROL-O / COSMOS Instances” on page 3-85.

Use this command to shut down all objects that are controlled by CONTROL-O / COSMOS in a system. This command sets the desired status of all CONTROL-O / COSMOS controlled objects to DOWN. The next time CONTROL-O / COSMOS scans the Object databases it notes the change and takes steps to ensure that CONTROL-O / COSMOS-controlled objects are brought down. The *COS*DOWN command invokes the COSDOWN rule that issues the following command:

```
COSCMD DESIDOWN ALL
```

*COS*OBJ — Display information about an Object Command

NOTE

As of version 6.1.00, this command uses a predefined prefix in place of the original “COS” prefix. For more information about these prefixes, see “Supporting Multiple CONTROL-O / COSMOS Instances” on page 3-85.

This command displays information about a specified object. The syntax for this command is:

```
COSOBJ obname
```

where *obname* is the name of the object to be described.

CONTROL-O / COSMOS searches all Working Object databases for the specified object. If more than one object with the specified name is located, all objects with the specified name are described in the output messages.

This command invokes the COSOBJ rule, which issues the following command:

```
COSCMD DISPALL ALL OBJECT obname
```

Supporting Multiple CONTROL-O/COSMOS Instances

Usually, only one CONTROL-O / COSMOS instance, running on one IOA installation, is started for each CPU. When more than one IOA installation is running on the same CPU, CONTROL-O / COSMOS normally runs on only one of these IOA installations. In these cases, CONTROL-O / COSMOS commands are handled by the single CONTROL-O / COSMOS instance.

There are, however, test and development environments that may require more than one CONTROL-O / COSMOS instance for each CPU. In such environments, it is essential that CONTROL-O / COSMOS commands are run on the correct instance of CONTROL-O / COSMOS.

CONTROL-O/COSMOS Command Prefix

Version 6.1.00 handles this situation by requiring you to define a 3-character command prefix for each CONTROL-O / COSMOS instance during ICE installation, using the PROCPRFO parameter of that instance. You use these defined prefixes in place of the regular COS prefixes for each CONTROL-O / COSMOS command.

Each CONTROL-O / COSMOS instance has a set of CONTROL-O / COSMOS rules beginning with the ??? prefix. These rules implement the CONTROL-O / COSMOS commands that use the defined command prefixes. For example, the rules ???INI and ???UP implement the commands that correspond to the *COS*INI and *COS*UP commands, respectively. These ???-prefixed rules check whether the prefix of a CONTROL-O / COSMOS command matches the prefix of the CONTROL-O / COSMOS instance specified in its PROCPRFO parameter.

When you enter a prefixed CONTROL-O / COSMOS command, all CONTROL-O / COSMOS instances in the IOA environment receive that command. Each instance compares the CONTROL-O / COSMOS command with the %%COSPREFIX variable, which was initialized with the value of the PROCPRFO parameter by the ON RULE USRCLEAR rule in the $COSMOSU table. If the defined prefix and command prefix do not match, command processing is terminated for that CONTROL-O / COSMOS instance. Otherwise, the command is executed normally. If the prefix does not match that of any CONTROL-O / COSMOS instances, the command is not executed.

When using the ??? prefix, note the following:

- If a rule with the ??? prefix triggers a CONTROL-O / COSMOS command, that command will use the same prefix as the triggering command.

- CONTROL-O / COSMOS commands are now trapped by the ON COMMAND rules that use the ??? prefix. These ON COMMAND rules check the prefix and, if there is a match, pass control to the corresponding ON RULE rules.

Example:

Two CONTROL-O / COSMOS instances are running on a CPU, for which you specified PR1 and PR2 as the prefixes, respectively. To run the *COS*INI command (described on page 77) on both instances, and the *COS*UP (described on page 84) on the second (PR2) instance, use the following set of commands:

```
PR1INI
PR2INI
PR2UP
```

Chapter 4

Implementation

This chapter includes the following topics:

Introduction

This chapter describes CONTROL-O / COSMOS implementation. It is highly recommended that you read this entire chapter before performing the tasks described below.

CONTROL-O/COSMOS Configuration

This section describes files used by CONTROL-O / COSMOS that must be modified to enable CONTROL-O / COSMOS to access relevant databases, and to trigger CONTROL-O / COSMOS rules.

RULELIST Member

This member (also known as the CONTROL-O Rule Table list) is a list of all rule tables to be loaded during CONTROL-O startup. This member is located in the CONTROL-O PARM library.

Verify that all CONTROL-O / COSMOS rule tables are listed in this member.

NOTE

Before verifying these rule tables, consider the following:

- If CONTROL-O / COSMOS was installed during CONTROL-O installation, the necessary rule tables are already referenced in the CONTROL-O Rule Table list (RULELIST member).
- If CONTROL-O / COSMOS was not installed during CONTROL-O installation, ensure that the CONTROL-O / COSMOS rule table is listed in the RULELIST member that is referenced in the CONTROL-O startup procedure.
- To use the COSMOS Sysplex support, load the CTOGATEI table, which supports communications, in addition to the COSMOS tables.

DAGLBLST Member

All CONTROL-O / COSMOS databases must be listed in the CONTROL-O Global Variable Pool list (member DAGLBLST). The following databases must be listed:

- All Object databases.
- All Method databases.
- The Prerequisite database.

Member DAGLBLST is located in the CONTROL-O PARM library. This member is checked during CONTROL-O startup for the names of all Global variable members and Variable databases. Listing CONTROL-O / COSMOS databases in this member ensures that the databases are loaded when CONTROL-O is started and that they are accessible to the AutoEdit facility (that is used to manage these databases).

NOTE

A series of Global variables are defined during CONTROL-O / COSMOS initialization. By default, these variables are stored in Global variable pool COSWORKV. Verify that this Global variable pool is listed in the DAGLBLST member.

If CONTROL-O / COSMOS was installed during the CONTROL-O installation process, CONTROL-O / COSMOS variable databases and the CONTROL-O / COSMOS Global variable pool COSWORKV may have already been added to this member.

This member must be changed when the XAE type 1 database is used in a Sysplex, as described in “Using CONTROL-O / COSMOS to Manage a Sysplex” on page 4-90.

For more information about Variable databases, see the online variable database definition discussion in the online facilities chapter of the *CONTROL-O User Guide*.

The AutoEdit facility is described in the AutoEdit facility chapter of the *CONTROL-O User Guide*.

Methods for managing variables in a variable database are described in the description of the DO SET parameter in the rule parameters chapter of the *CONTROL-O User Guide*.

CONTROL-O Variable Database Files

CONTROL-O / COSMOS databases are stored together with all other variable databases in the CONTROL-O variable database files. Verify that the size specified for these files is sufficient for expected CONTROL-O / COSMOS databases.

For more information, see the CONTROL-O chapter of the *INCONTROL for z/OS Administrator Guide*.

Using CONTROL-O/COSMOS to Manage a Sysplex

You can use XAE type 1 databases with the CONTROL-O / COSMOS Working Object Databases. With XAE type 1 databases, users that are logged on to one system can see the objects of different Sysplex participants, and can manage them from a single screen. For example, a user logged on to SYSTEM A can start or stop an object running on SYSTEM B.

To enable CONTROL-O / COSMOS to manage a Sysplex, each line in member DAGLBLST that defines a working database must be changed (for example, from DBTEMP to S1TEMP). In addition, the CTOGATEI table must be loaded from the RULELIST member in the CTO*prefix*.RULES library.

NOTE

When you use CONTROL-O / COSMOS to manage a Sysplex, if you are defining CPU names within member COSMOLST, (for more information, see "COSMOLST Member" on page 90), you must use the exact name of each CPU's corresponding system name as defined in the Sysplex environment.

COSMOLST Member

The COSMOLST member in the CONTROL-O PARM library contains basic information about all Object databases to be managed by CONTROL-O / COSMOS. The information in the COSMOLST member is used by:

- CONTROL-O / COSMOS at initialization, to determine information about the working environment.
- CONTROL-O / COSMOS initialization rules, described in Chapter 3, "Component Details," to determine values for Global AutoEdit variables used by CONTROL-O / COSMOS rules.
- CONTROL-O / COSMOS to generate the CONTROL-O / COSMOS Database Status screen.

A sample COSMOLST member is shown below:

Figure 16 Sample COSMOLST

```
H FREE     COSALLPR 00000000 000000000
T COSSTCSD COSSTCOB COSALLMT FREE     UP       DOWN     UNKNOWN
T COSVTMSD COSVTMOB COSALLMT FREE     ACTIVE   INACTIVE UNKNOWN
T COSIMGSD COSIMGOB COSALLMT FORCE_OK UP       DOWN     UNKNOWN
C ESA1     ESA2     ESA3     ESA4     ESA5     ESA6     ESA7
```

Format of the COSMOLST Member

The first column of each record in member COSMOLST indicates the type of record in the line. Valid values are:

- H – Header record
- T – Object Database record
- C – CPU record

Valid format for COSMOLST records of each type are described below.

Header Record

Only one header record can be specified. It contains general information about CONTROL-O / COSMOS.

Table 24 Header Record

Columns	Description
03-10	Default system wide CONTROL-O / COSMOS operating mode at startup. Valid values are FREE, HELD, FORCE_OK and NOPRE.
12-19	Name of the CONTROL-O / COSMOS Prerequisite database.
21-28	Operational flags. For future use (now all zeroes).
30-37	CONTROL-O / COSMOS flags. If set to Y, a COSFSMPR message is issued for each missing prerequisite.

Object Database Record

One or more database records can be specified. They contain information related to each Object database.

CPU Record

Table 25 CPU Record Description

Columns	Description
03-10	Name of the Source Object database.
12-19	Name of the Working Object database.
21-28	Name of the Method database associated with this Object database.
30-37	Default mode of the database at startup. valid values: FREE, HELD, FORCE_OK, and NOPRE.
39-46	Name of the UP status (for example, UP, ONLINE or ACTIVE) for objects in the Object database.
48-55	Name of the DOWN status (for example, DOWN, OFFLINE or INACTIVE) for objects in the Object database.
57-63	Name of the UNKNOWN status (for example, UNKNOWN or NOTFOUND) for objects in the Object database.

CPU records are used to indicate the system name or the SMF ID of CPUs in your environment. Each CPU name specified in a CPU record is stored in a variable when CONTROL-O / COSMOS is activated. These variables can be referenced at your convenience using CONTROL-O rules.

NOTE

When using CONTROL-O / COSMOS to manage a Sysplex, as described on page 90, if you want to be able to browse several systems from the online screen, you must define each CPU name using its corresponding system name, as defined in the Sysplex environment.

Table 26 CPU Name Definitions

Column	CPU
03-10	CPU name 1
12-19	CPU name 2
21-28	CPU name 3
30-37	CPU name 4
39-46	CPU name 5
48-55	CPU name 6
57-63	CPU name 7

One or more CPU records can be specified.

Testing CONTROL-O/COSMOS Operations

A set of supplied demo CONTROL-O / COSMOS databases can be used to manage a set of simulated CONTROL-O / COSMOS objects. Actions performed with these objects cause actual messages to be issued, but are used to manage simulated objects, not the real objects in your computing environment.

Use the CONTROL-O / COSMOS demo environment to familiarize yourself with CONTROL-O / COSMOS commands and options without affecting your system, and to test modifications to CONTROL-O / COSMOS rules before implementing them in the production environment.

NOTE

Simulated objects in the CONTROL-O / COSMOS demo environment issue messages that are identical to certain messages issued by true objects in your production environment. To avoid inadvertently affecting objects in your production environment, if you already have automation rules in use at your site that respond to these messages (for example, startup messages), ensure that these automation rules are disabled while you are working in the CONTROL-O / COSMOS demo environment.

Components

The following components are supplied as part of the CONTROL-O / COSMOS demo environment:

- procedures that simulate messages issued by common started tasks
- object database describing started tasks

  Actions on objects in this database simulate actions on real started tasks. Starting or stopping a demo started task produces the same messages as starting or stopping a real started task without actually starting or stopping the real started task.

  For example, procedure *xxx*JES2 (the demo JES2 object) issues the same messages that JES2 issues when it is started or stopped. IPL or shutdown can be simulated without the need to initialize or shut down the computer.

  To enable running several demo systems in the same CPU the demonstration database object name prefixes are changed, in the COSINI command code, from CTO to the PROCPRFO prefix.

  For example, if the PROCPRFO prefix is “O60,” an object named CTOJES2 is renamed to O60JES2.

- Object database containing demo VTAM LUs
- Prerequisite database describing interdependencies of objects in the demo Object databases

  Empty Object database (only the columns are defined) used to store started task objects defined by the SYSIMAGE facility. This database helps demonstrate how CONTROL-O / COSMOS handles your production environment.

- Demo Method database

  This Method database contains descriptions of some of the methods required to manage objects in the Object databases of the demo environment.

CONTROL-O / COSMOS rules of all types. CONTROL-O / COSMOS rules are described in Chapter 3, “Component Details.”

Method rules supplied with CONTROL-O / COSMOS demonstrate basic management of CONTROL-O / COSMOS-controlled objects. These rules can later be used as a basis for creation of additional rules used to manage objects in your production environment.

Demo Databases Supplied With CONTROL-O/COSMOS

The following demo databases are provided:

Table 27 Demo Databases

Database	Description
COSALLPR	Demo Prerequisite database.
COSALLMT	Demo Method database.
COSSTCOB	Demo Working Object database for started task objects.
COSVTMOB	Demo Working Object database for VTAM objects.
COSIMGOB	Demo Working Object database for objects defined using the SYSIMAGE facility, described on page 96.
	Note: When the input file for the SYSIMAGE facility is properly customized, modify the SYSIMAGE rules to build objects in the appropriate production Object database.
COSSTCSD	Demo Source Object database for started task objects.
COSVTMSD	Demo Source Object database for VTAM objects.
COSIMGSD	Demo Source Object database for objects defined using the SYSIMAGE facility, described on page 96.

Working With the CONTROL-O/COSMOS Demo Environment

You can use the CONTROL-O / COSMOS demo environment to familiarize yourself with CONTROL-O / COSMOS functions (screens, commands, and so on). Use the following steps to guide you through CONTROL-O / COSMOS.

1. Start CONTROL-O (if it is not already active).

2. Start CONTROL-O / COSMOS using command *xxx*INI (if this command was not already issued by a rule at CONTROL-O / COSMOS startup).

3. Enter the CONTROL-O / COSMOS Object Status screen using option OBJECT in the CONTROL-O / COSMOS Main menu (screen OC). You can use this screen to view the current status of CONTROL-O / COSMOS-controlled objects.

4. Exit the Object Status screen, and issue operator command *xxx*UP. This command sets the mode of demo Object databases to FREE and specifies UP as the desired status for each object.

5. Return to the Object Status screen to observe how CONTROL-O / COSMOS-controlled objects are handled as a result of command *xxx*UP.

   CONTROL-O / COSMOS-controlled objects are brought up in the order dictated by prerequisites defined in the demo Prerequisite database (for example, *xxx*JES2 is brought up before *xxx*VTAM, followed by applications that require VTAM to be up).

6. When all CONTROL-O / COSMOS-controlled objects have a status of STEADY UP, issue command *xxx*DOWN.

   Objects are brought down in the reverse order from which they were brought up. This is due to inverse application of CONTROL-O / COSMOS prerequisites.

7. Use the SYSIMAGE facility, described on page 96, to build an Object database that reflects your production environment.

   Objects defined using the SYSIMAGE facility are assigned a mode of FORCE_OK. This ensures that objects in your production environment are not modified by CONTROL-O / COSMOS while operating in a demo environment.

   For more information about the SYSIMAGE facility, see “SYSIMAGE Facility – Automatic Generation of an Object Database” on page 96, and the CTOCTI utility in the *INCONTROL for z/OS Utilities Guide.*

8. Experiment with options in the Object Status screen and the Database Status screen. For more information about these screens, see Chapter 2, “Online Facilities.”

9. Review the rules supplied with CONTROL-O / COSMOS, and familiarize yourself with their logic.

SYSIMAGE Facility – Automatic Generation of an Object Database

The SYSIMAGE facility is used to automatically generate an object database that reflects your working environment. Table 28 describes the components that make up the SYSIMAGE facility:

Table 28 SYSIMAGE Facility Components

Item	Description
CTOCTI utility	This utility, located in the JCL library, scans the system for started tasks, and issues messages with started task information. An input file for the CTOCTI utility is used to specify information about started tasks in your environment. Ensure that the information in this file is as complete and accurate as possible, so that the automatically generated object database contains an accurate description of your environment. For more information about the CTOCTI utility, see the *INCONTROL for z/OS Utilities Guide.*
CTOCTA utility	This utility, located in the JCL library, provides the names of programs to be executed from a specified address space. This helps determine the information to be included in the input file for the CTOCTI utility. For more information about the CTOCTA utility, see the *INCONTROL for z/OS Utilities Guide.*
CTO659I rule	A CONTROL-O rule triggered by message CTO659I, issued by the CTOCTI utility. This rule creates an object database row for each occurrence of the message, according to information in the message. By default, the rule inserts rows in object database COSIMGSD. This object database is defined in the DAGLBLST member with an attribute of DBINPUT, which means that no changes can be made to this database. This rule can be customized, for example, to automatically create rows in the prerequisite database or method definitions in a method database.

To implement the SYSIMAGE facility, perform the following steps:

1 Run the CTOCTI utility to identify started tasks that were not defined in the input file to the utility.

2 If the CTOCTI utility reports undefined tasks, do the following:

   A Run the CTOCTA utility to determine which information is inaccurate or missing from the input file.

   B Update the input file.

   C Rerun the CTOCTI utility.

   D Repeat these steps until the desired tasks are defined in the input file.

3 When you are satisfied with the contents of the input file, do one of the following:

   - Change the attribute of object database COSIMGSD, located in member DAGLBLST, to DBINOUT. Reload the object database using the LOADGLOBAL command, described in the CONTROL-O chapter of the *INCONTROL for z/OS Administrator Guide*.

   - Modify the CTO659I rule to point to a different object database that has an attribute of DBINOUT.

   Rows reflecting the objects (resources) in your system are added to a specified object database in memory. Use the WRITEGLOBAL command to update the object database in the CONTROL-O Variable Database files. For more information, see the CONTROL-O chapter of the *INCONTROL for z/OS Administrator Guide*.

4 Rerun the CTOCTI utility.

   For more information about utilities CTOCTA and CTOCTI, see the *INCONTROL for z/OS Utilities Guide*. For more information about valid attributes for variable databases, see the CONTROL-O chapter of the *INCONTROL for z/OS Administrator Guide*.

Implementing CONTROL-O/COSMOS in a Production Environment

CONTROL-O / COSMOS is implemented in a production environment after you have experimented with all aspects of CONTROL-O / COSMOS in the demo environment and feel comfortable with the logic used to implement CONTROL-O / COSMOS functions.

Use the following steps to implement CONTROL-O / COSMOS in a production environment:

1. Fill CONTROL-O / COSMOS databases for the production environment. Skeleton database definitions (meaning databases with only columns defined) are provided with CONTROL-O / COSMOS.

   The demo databases provided with CONTROL-O / COSMOS can be used as examples for the type of information that is inserted in the production databases.

   The SYSIMAGE facility can be used to generate an object database describing objects in your production environment. Modify SYSIMAGE rules CTO658I and CTO659I to point to supplied production databases PRDSTCOB and PRDSTCSD.

   For more information, see "SYSIMAGE Facility – Automatic Generation of an Object Database" on page 96.

2. Change the following Global variable specifications in rule COSCLEAR:

   Replace %%COSSTC = COSSTCOB with %%COSSTC = PRDSTCOB

   Replace %%COSVTM = COSVTMOB with %%COSVTM = PRDVTMOB

   This ensures that all CONTROL-O / COSMOS-provided sample methods and rules use the production databases instead of the demonstration databases. (The provided rules obtain the database names from the AutoEdit variables when necessary.)

3. Modify member COSMOLST so that it references CONTROL-O / COSMOS production databases. Sample content for a production COSMOLST member is provided in member COSMOPRD in the CONTROL-O PARM library. Rename this member to COSMOLST, or modify the existing COSMOLST member.

The first time CONTROL-O / COSMOS is activated in a production environment, it is recommended that you first initialize all CONTROL-O / COSMOS objects with a mode of FORCE_OK. This enables you to transfer control of specific objects to CONTROL-O / COSMOS control one at a time, enabling you to more easily observe and customize the way CONTROL-O / COSMOS handles objects at your site.

Production Databases Supplied With CONTROL-O/COSMOS

The following production databases are provided:

Table 29 Production Databases

Database	Description
PRDALLPR	Prerequisite database.
PRDALLMT	Method database.
PRDSTCOB	Working Object database for started task objects.
PRDVTMOB	Working Object database for VTAM objects.
PRDSTCSD	Source Object database for started task objects.
PRDVTMSD	Source Object database for VTAM objects.

Index

Notes

62169